Bringing Image Structure to Video Transformers via Frame-Clip Consistency of Object Tokens

Anonymous Author(s)
Affiliation
Address
email

Abstract

Recent action recognition models have achieved impressive results by integrating objects, their locations and interactions. However, obtaining dense structured annotations for each frame is tedious and time-consuming, making these methods expensive to train and less scalable. At the same time, if a small set of annotated images is available, either within or outside the domain of interest, how could we leverage these for a video downstream task? We propose a learning framework StructureViT (SViT for short), which demonstrates how utilizing the structure of a small number of images only available during training can improve a video model. SViT relies on two key insights. First, as both images and videos contain structured information, we enrich a transformer model with a set of *object tokens* that can be used across images and videos. Second, the scene representations of individual frames in video should "align" with those of still images. This is achieved via a *Frame-Clip Consistency* loss, which ensures the flow of structured information between images and videos. We explore a particular instantiation of scene structure, namely a *Hand-Object Graph*, consisting of hands and objects with their locations as nodes, and physical relations of contact/no-contact as edges. SViT shows strong performance improvements on multiple video understanding tasks and datasets.

1 Introduction

Semantic understanding of videos is a key challenge for machine vision and artificial intelligence. It is intuitive that video models should benefit from incorporating scene structure, e.g., the objects that appear in a video, their attributes, and the way they interact. Indeed, several works [5, 25, 26, 33, 34, 41, 57, 59, 62, 74, 79, 80, 82, 88] have demonstrated that incorporating structured representations into models improves both performance and sample efficiency. In particular, [33, 80, 88] showed how to incorporate structured representations by utilizing static image object annotations in videos.

Recently, vision transformers (ViT) [20] have emerged as the leading model for many vision applications [4, 21, 12]. This begs the question, how can structured representations be used to leverage video transformers for video tasks? Past works [4, 21] have proposed video transformer models but these approaches were not focused on learning structured representations. A recent line of works [33, 80, 88] utilized objects for structured approaches. [82] leveraged the decoder with CNNs region crop features from SlowFast [22], and [33] proposed a new block that incorporates object representations. However, these approaches are limited because they use external pretrained detectors for the frames, which require additional processing time (also during inference), they ignore the potential benefits of the connection between the detection and the other tasks, and it is not clear how to use them with more detailed structured annotations. Finally, [88] proposed a structured model for action detection that explicitly utilizes image annotations, but not in a transformer architecture. Our approach, described next, builds on these prior works and proposes an effective way to leverage scene structure in video transformers, using static image supervision.

Most structured transformer-based models for videos require structured annotations during training. For example, frame-level object bounding boxes are considered available in [5, 7, 33, 34, 58, 80].

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

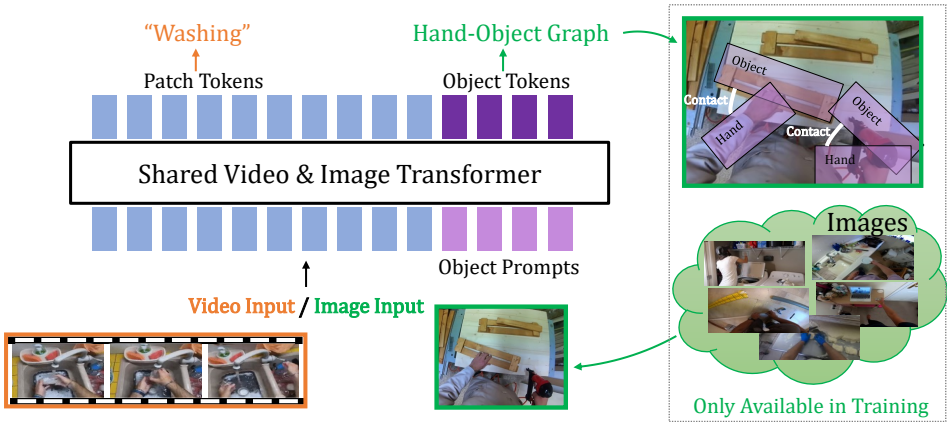

Figure 1: Our approach SViT brings structured scene representations from still images into video. We use the **HA**nd-**O**bject **G**raph (HAOG) annotations of still images that are only available during training, and videos, which may come from a different domain, with their downstream task annotations. We design a shared video & image transformer that can handle both video and image inputs. During training, given an image, the patch and object tokens are processed together, and the object prompts learn to predict the HAOG, whereas given a video input, the transformer predicts the video label (e.g. *Washing*) based on the patch tokens. During inference, the learned object prompts, which have captured the structured scene information from images, are used to predict the video label.

When it comes to structured annotations of static images, there are numerous datasets with annotations such as boxes, visual relationships, attributes, and segmentations [10, 48, 58]. Since we cannot always expect to have annotated images that perfectly align with our downstream video task, how can these resources be used to build better transformer-based video models? In this work, we propose such an approach, which is particularly designed for transformer architectures and offers an effective mechanism to leverage image structure to improve video transformers.

A natural first step towards image-video knowledge sharing is to use the same transformer model to process both: namely, to view images as "single frame videos", and feed them to a video transformer model. However, this still leaves two key questions: how to model structured information, and how to account for domain mismatch between images and videos. Towards this end, we introduce two key concepts: (i) A set of transformer "object-tokens" – additional tokens initialized from learned embeddings (we refer to these embeddings as "object prompts") that are meant to capture object-centric information in both still image and video. These tokens can be supervised based on the scene structure of images, and that information will also be propagated to videos. To formalize the image structure, we proposed the HAnd-Object Graph (HAOG), a simple representation of the interactions between hands and objects in the scene. (ii) A novel "Frame-Clip Consistency loss" that ensures consistency between the "object-tokens" when they are part of a video vs. a still image. Moreover, as we show in our experiments, our approach does not require any alignment between object categories appearing in still images and the ones appearing in the videos of the target downstream task. We name our proposed approach StructureViT (*Bringing Scene **Structure** from Images to **Vi**deo via Frame-Clip Consistency of Object **T**okens*, or SViT for short). See Figure 1 for an overview.

Prior work has explored several other video & image models. The I3D model [13] was one of the first successful attempts to leverage ImageNet architectures in video. They proposed to simply convert image 2D classification models into 3D ConvNets by inflating all the filters and pooling kernels. Here, our focus is on learning the shared structured representations between the two domains in order to transfer this knowledge into videos. Several other methods have explored simultaneous image and video training in the context of multi-task [6] and multi-modal [27] learning. However, these works did not utilize structural information, nor did they promote consistency between image and video predictions – which, as we show, leads to significant gains in downstream video task performance.

To summarize, our main contributions are as follows: (i) we propose a new method for exploiting structured information present in images in order to improve performance of video understanding tasks; (ii) we propose a novel concept of "object tokens" - a form of learned prompts to a video & image transformer, which during training are associated to the available structure representation of the still images (HAOG). (iii) we introduce a new Frame-Clip Consistency loss, which promotes consistency between image-level and video-level predictions made by a shared transformer backbone. We show how this loss helps to drive performance improvements in downstream video tasks even in cases when image labels are unrelated to the video task; (iv) we demonstrate improved performance on several video understanding benchmarks, highlighting the effectiveness of the proposed approach.

2 The SViT Approach

Our approach SViT learns structured representation that is exhibited in both static images and video. We consider the setting where the main goal is to learn video understanding downstream tasks while leveraging structured image data. In this paper, we focus on the following video tasks: action recognition and object state classification & localization. In training, we assume that we have access to task-specific video labels and structured scene annotations for images, *Hand-Object Graph*s. Based on the structured representations obtained by using the *object tokens* and regularized by the *Frame-Clip Consistency loss*, the inference is performed only for the downstream video task without explicitly using the structured image annotations.

Further, we begin by describing the Hand-Object Graph (HAOG) (Section 2.1), then introduce our SViT model (Section 2.2), and the Frame-Clip Consistency loss function (Section 2.3). Our method is schematically illustrated in Figure 2.

2.1 The Hand-Object Graph

As mentioned above, our motivation is to bring scene structure from still images into video. In order to achieve this, one component of this work is the construction of a semantic representation of the interactions between the hands and objects in the scene. Specifically, we propose to use a graph-structure we call *Hand-Object Graph* (HAOG), see Figure 1. The nodes of the HAOG represent two hands and two objects with their locations, whereas the edges correspond to physical properties such as contact/no contact. Formally, an HAOG is a tuple (O, E) defined as follows:

Nodes O: A set O of n objects. Human-object interaction scenes, which we focus on in this work, usually consist of up to two hands and objects. Thus our node categories are "hand" and "object"; they may also contain attributes (e.g., spatial coordinates). We assume that each image comes with the bounding boxes describing the locations of the two hands, and the objects they interact with. The bounding boxes for the left and right hand are denoted as $\boldsymbol{b}_1, \boldsymbol{b}_2 \in \mathbb{R}^4$, respectively, while for the two corresponding objects interacting with the hands are denoted as $\boldsymbol{b}_3, \boldsymbol{b}_4$. We also assume access to four binary variables $e_1, \ldots, e_4 \in \{0, 1\}$ that indicate whether the corresponding hands or objects in fact exist in the image (e.g., if $e_1 = 0$ the left hand is not in the image, and $\boldsymbol{b}_1$ should be ignored).

Edges E: The edges are represented as labeled directed edges between hand and object nodes. Edge vocabulary describes the physical properties of the scene. In our current modeling, each edge is characterized by a physical property of "contact" or "no-contact". We denote two binary variables $c_1, c_2 \in \{0, 1\}$ that specify if each of the hands is in direct contact with the corresponding object (the left hand $\boldsymbol{b}_1$ can only interact with the object $\boldsymbol{b}_3$, similar for the right hand). In principle, we can also include properties such as distances between objects, 3D hand poses, etc., where available.

We note that images annotated consistent with our defined HAOG format can be found in several datasets [31, 58, 71], and we leverage these in our experiments.

2.2 The SViT Transformer Model

We next present the SViT model: a video transformer that employs *object tokens* as a means to capture structure in both images and videos. We begin by reviewing the video transformer architecture, which our model extends. Next, we explain how we process both images and videos. Finally, we describe object tokens and how they are used in videos and still images.

Shared Video & Image Transformer. Video transformers [4, 11, 21, 33] extend the Vision Transformer model to the temporal domain. Similar to vision transformers, the input is first "patchified" but with temporally extended 3-D convolution (instead of 2-D) producing a down-sampled tensor X of size $T \times H \times W \times d$. We refer to this as "patch tokens". Then, spatio-temporal position embeddings are added for providing location and time information. Next, multiple stacked self-attention blocks are applied repeatedly on the down-sampled tensor X to produce the final vector representation using mean pooling over the patch tokens.

In our approach, we want to be able to process batches of images or videos. A key desideratum in this context is to be able to input both videos and still-images into the same model. Towards this end, our first change is to rely only on 2-D convolutions in the first transformer step instead of 3-D. This allows single images to be used as inputs without padding, and does not hurt performance in practice.

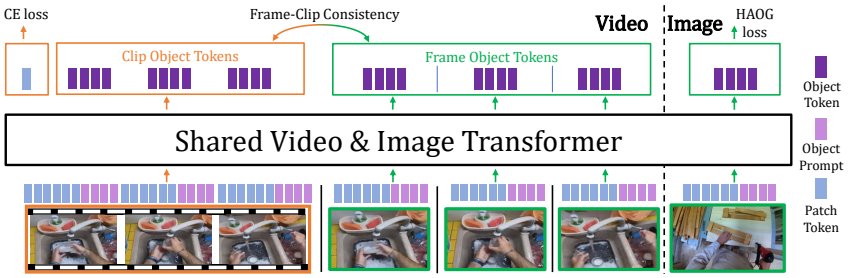

Figure 2: Our shared video & image transformer model processes two different types of tokens: standard patch tokens from the images and videos (**blue**) and the object prompts (**purple**), that are transformed into object tokens (**purple**) in the output. During training, the object tokens (**purple**) are trained to predict the HAOG for still images. For video frames that have no HAOG annotation, we use our "Frame-Clip" loss to ensure consistency between the "frame object tokens" (resulting from processing the frames separately) and the "clip object tokens" (resulting from processing the frame as part of the video). Last, the final video downstream task prediction results from applying a video downstream task head on the average of the patch tokens in the transformer output (after they have interacted with the clip object tokens (**purple**)).

Object tokens. Our goal is to learn a shared structured representation between the video and the image domains. We do this by adding transformer tokens to represent objects. These tokens are functionally similar to the patch tokens, with two exceptions. First, they are initialized with additional trained embeddings, which we referred to as "object prompts". We consider the "object prompts" as a bridge between the images and the videos since they are being used in prediction tasks across both domains. Second, they are used to predict the structured representations when those are available (here, we use the Hand-Object Graphs structure for still images). An object token is used to predict properties of the corresponding node in the HAOG, and the concatenation of two object tokens is used to predict edge properties.

Next, we describe the model more formally. Let n be the maximum number of modeled objects per frame (or still image). We have n tokens for each frame, and thus a total of $T \times n$ tokens. The token for object i at frame t is initialized with the vector $\boldsymbol{o}_i + \boldsymbol{r}_t$ where $\boldsymbol{o}_i \in \mathbb{R}^d$ is a learned object prompt and $\boldsymbol{r}_t \in \mathbb{R}^d$ is a learned temporal positional embedding (the same one used for initializing the patch tokens). With these new tokens, we have $T \times (H \times W + n)$ tokens (i.e, vectors in $\mathbb{R}^d$), and these together will go through the standard self-attention layers.

We denote the operator that outputs the final representation of the object tokens by F_O, where for a video V we have $F_O(V) \in \mathbb{R}^{T \times n \times d}$ and for an image I we have $F_O(I) \in \mathbb{R}^{n \times d}$. The $F_O(I)$ representation is used to predict structured representations for single images. We also include a loss that makes single frame representations align with those of clip representations (see Section 2.3). The operator that outputs the final representation vector used for the video downstream task is denoted by F_{CLS}, where for video V we have $F_{CLS}(V) \in \mathbb{R}^d$.

Finally, our method can be used on top of the most common video transformers (MViT [21], TimeSformer [11], Mformer [65], Video Swin [55]). For our experiments, we use the MViTv2 [52] model because it performs well empirically.

2.3 Losses and Training

During training we have a set of images annotated with HAOGs and videos with downstream task labels. We use these as inputs to our model (Section 2.2) and optimize the losses described below.

Video loss. Each training video V has a corresponding ground-truth category $Y \in \{1, \ldots, K\}$. As noted above, we have a vector representation for the entire video denoted by $F_{CLS}(V)$. Thus, we simply use a neural network to predict logits for Y. Specifically, our predicted logits are: $\hat{Y} = FC(F_{CLS}(V)) \in \mathbb{R}^K$, where FC is a trainable fully connected network. We then consider a standard cross-entropy between the predicted logits $\hat{Y}$ and true labels Y.

$$\mathcal{L}_{Vid} := \text{CE}(\text{Softmax}(\widehat{Y}), Y) \ , \tag{1}$$

HAOG loss. The transformer outputs a set of n object tokens for each image. Since we have supervision for these objects, we predict structured representations from the tokens, and introduce a loss to optimize these predictions. Recall our notation from Section 2.1. We let the transformer use $n = 4$ object tokens which correspond to the two hands (bounding boxes $\boldsymbol{b}_1, \boldsymbol{b}_2$) and two manipulated

objects (bounding boxes $\boldsymbol{b}_3, \boldsymbol{b}_4$). The jth token is used to predict a corresponding bounding box b_j, as well as the existence variable e_j. Formally, we let:

$$\hat{\boldsymbol{b}}_j = \text{FC}_{bb}(F_O^j(I)) \ , \ \hat{e}_j = \text{FC}_e(F_O^j(I)) \ , \tag{2}$$

where FC_{bb}, FC_e are fully connected networks with four and one output respectively, $F_O^j(I)$ is the jth object token, and I is the input image. Formally, this define the loss of the nodes as following:

$$\mathcal{L}_{Nodes} = \sum_{i=1}^{4} \text{BCE}(\text{Sigmoid}(\hat{e}_i), e_i) + e_i \left(\text{GIoU}(\hat{\boldsymbol{b}}_i, \boldsymbol{b}_i) + L_1(\hat{\boldsymbol{b}}_i, \boldsymbol{b}_i) \right) \tag{3}$$

where L1 is the standard L1 distance, and GIoU is used as in [68]. Furthermore, we predict the contact variable from the concatenation of the corresponding hand and object tokens. Namely for $j \in \{1, 2\}$:

$$\hat{c}_j = \text{Softmax}(\text{FC}_c([F_O^j(I), F_O^{j+2}(I)])) \tag{4}$$

where FC_c is a fully connected with one output. Thus, we define the loss of the edges as follows:

$$\mathcal{L}_{Edges} = \sum_{i=1}^{2} \text{CE}(\hat{c}_i, c_i) \tag{5}$$

Finally, to obtain the loss, we compare the predictions and ground-truth labels as follows:

$$\mathcal{L}_{HAOG} = \mathcal{L}_{Nodes} + \mathcal{L}_{Edges} \tag{6}$$

Frame-Clip Consistency Loss. Since we have different losses for images and video, the model could find a way to minimize the image loss in a way that only applies to the images and does not transfer to video.[1] To avoid this, we need to make sure that the video representation contains the same type of information as the still images, and in particular the structured information carried by the object tokens. To achieve this, we explicitly enforce the object tokens to be consistent across still images (frames) and videos (clips) using a frame-clip Consistency loss. To calculate the loss, we process each video twice: once as a clip, and once as a batch of T frames. Namely, a clip V consisting of T frames $I_1, ..., I_T$, will be processed once as $F_O(V)$ and once as a list of images $(F_O(I_1), ..., F_O(I_T))$. This results in two groups of $T \times n$ object tokens, the first are the *clip object tokens*, denoted as, $\mathcal{O}_{clip} := F_O(V) \in \mathbb{R}^{T \times n \times d}$, the second are the *frame object tokens*, denoted as $\mathcal{O}_{frames} := [F_O(v_1), ..., F_O(v_T)] \in \mathbb{R}^{T \times n \times d}$. Since both groups originate from the same input, each clip object token has a single corresponding frame object token. Namely, there is an alignment between the clip object tokens and the frame object tokens. Following the intuition that each clip object token should contain the information from its corresponding frame object token and potentially more (the temporal context), we minimize the $L1$ distance between each clip object token and frame object token, after applying a fully-connected layer to each clip object token. Namely:

$$\mathcal{L}_{Con} := L_1(\text{FC}(\mathcal{O}_{\text{clip}}), \mathcal{O}_{\text{frames}}) \tag{7}$$

Overall loss. The total loss consists of $\mathcal{L}_{Con}$, $\mathcal{L}_{HAOG}$ and $\mathcal{L}_{Vid}$, where $\mathcal{L}_{Vid}$ is the loss of the main task. Each of the three terms in the loss is multiplied by a hyper-parameter (λ_{Con}, λ_{HAOG}, λ_{Vid}), and the total loss is the weighted combination of the three terms:

$$\mathcal{L}_{\text{Total}} := \lambda_{Con}\mathcal{L}_{Con} + \lambda_{HAOG}\mathcal{L}_{HAOG} + \lambda_{Vid}\mathcal{L}_{Vid} \tag{8}$$

3 Experiments and Results

We begin by describing the datasets, implementation details, and baselines and variants. We then evaluate our approach on several benchmarks for video understanding. Specifically, we consider the following tasks: Compositional Action Recognition (Section 3.4), Object State Change Classification & Localization (Section 3.5), and Action Recognition (Section 3.6).

[1] For example, it could "overfit" to relying on the positional embedding of the first frame only when minimizing image-related losses.

3.1 Datasets

We first describe the datasets used for the downstream video tasks, and those used as "auxiliary" datasets with annotated images. We use the following video datasets: **(1) Something-Something v2 (SSv2)** [30] is a dataset containing 174 action categories of common human-object interactions. **(2) SomethingElse [58]** which exploits the compositional structure of SSv2, where a combination of a verb and a noun defines an action. We follow the official compositional split from [58], which assumes the set of noun-verb pairs available for training is disjoint from the set given at test time. **(3) Ego4D** [31] is a new large-scale dataset of more than 3,670 hours of video data, capturing the daily-life scenarios of more than 900 unique individuals from nine different countries around the world. **(4) Diving48** [53] contains 48 fine-grained categories of diving activities. For "auxiliary" image datasets we use the Ego4D, and the **100 Days of Hands (100DOH)** [71] datasets. We collected 79,921 annotated images from 100DOH, and 57,213 annotated images from Ego4D.[2] The collected object boxes from SSv2, 100DOH, and Ego4D are purely human annotated. In SSv2 and Ego4D, contact relations between the object and hand are not annotated, so we assign the closest object to the hand for each hand. The contact relations for 100DOH are available in the dataset.

For the image datasets, the image annotations are based on frames selected from videos, but these are often sparsely selected from the videos, so it is natural to treat them as annotated still images.[3]

3.2 Implementation Details

SViT is implemented in PyTorch, we include the code in the supplementary and will release it upon acceptance. Our training recipes and code are based on the MViTv2 model, and were taken from `https://github.com/facebookresearch/mvit`. We pretrain the SViT model on the K400 [44] *video* dataset with our *auxiliary image* datasets. Then, we finetune on the target video understanding task (detailed in Section 3.1) together with the *auxiliary image* datasets and the SViT loss. Each training batch contains 64 images and 64 videos in order to minimize the overall loss in Equation 8.

3.3 Baselines and SViT variants

Baselines. In the experiments, we compare SViT to several models from prior work reported for the corresponding datasets. These include the following approaches: *I3D* [13], *SlowFast* [22], and the state-of-the-art transformers – *Mformer* [65] and *MViTv2* [52]. In addition, we provide the *MViTv2 multi-task (MViTv2 MT)* baseline for comparing a naive multi-tasking approach to our model. This is a version of MViTv2 which performs video classification and HAOG prediction from images. The MViTv2 model is enhanced with two heads: the first head is composed of an MLP that predicts the video labels given a final representation vector, the second head consists of MLPs that predict the HAOG information (i.e., hands and object boxes, and contact information).

SViT Variants. As mentioned above, we use several sources of auxiliary image datasets. In some cases, we have auxiliary images that are frames from video understanding datasets. For example, in Ego4D, we have both videos and frames annotated from some of these videos. In this sense, the auxiliary data is "in-domain" for some of the video tasks. In order to evaluate the importance of this effect, we explore "Different Domain" (*SViT-DD*) training, which avoids using images from a given video task as auxiliary during finetuning and pretraining. Similarly, "In Domain" (*SViT-ID*) refers to the case where we use only images from the video task during finetuning, but not necessarily during pretraining . We consider another variant that does not use the auxiliary images during finetuning at all (but does use them during pre-training), and refer to it as *SViT-SFT*. Finally, to explore the effect of data-size in the In Domain setting, we evaluate a version *SViT* $\times X\%$ that only uses X% of the in-domain finetuning data. For MViTv2 MT, the auxiliary images correspond to the "In Domain" setting (i.e., same setting as for SViT-ID).

Auxiliary Images datasets. For SViT-DD, we generally use 100DOH and Ego4D for pretraining and finetuning, while for the Ego4D tasks, we only use 100DOH. SViT-ID uses 100DOH and Ego4D for pretraining and in-domain downstream frames for finetuning. The SViT-SFT pretraining setting is similar to that of SViT-DD. SViT $\times$X% setting is similar to that of SViT-ID.

[2] Comparatively, SomthingElse contains 2,550,700 frames, showing that image datasets are relatively small.

[3] The SomethingElse/SSv2 dataset has annotations for all clip frames, but we show in the experiments that SViT only needs a small percentage of these.

Model	Boxes Annotations	Compositional		Base		Few-Shot	
		Top-1	Top-5	Top-1	Top-5	5-Shot	10-Shot
SlowFast [22]	✗	45.2	73.4	76.1	93.4	22.4	29.2
TimeSformer [11]	✗	44.2	76.8	79.5	95.6	24.6	33.8
Mformer [65]	✗	60.2	85.8	82.8	96.2	28.9	33.8
MViTv2 [52]	✗	63.3	87.5	83.7	96.8	32.7	40.2
MViTv2 MT	✓	63.8	87.3	85.2	97.1	33.7	41.4
SViT-SFT	✓	64.2	87.9	85.0	97.1	33.8	41.7
SViT-DD	✓	65.1	88.0	85.8	97.4	34.6	42.5
SViT $\times 2\%$	✓	65.6	88.1	85.9	97.4	34.4	43.1
SViT-ID	✓	65.8	88.3	85.6	97.3	34.4	42.6

Table 1: **Compositional and Few-Shot Action Recognition** on the "SomethingElse" dataset.

(a) Object State Change in Ego4D.

Model	Temporal loc. error	Cls. top-1
Bi-LSTM †	0.790	65.3
BMN†	0.780	-
I3D ResNet-50†	0.739	68.7
MViT-v2†	0.702	71.6
MViTv2 MT	0.595	71.1
SViT-SFT	0.544 (−.158)	70.4 (−1.2)
SViT-DD	0.527 (−.175)	73.6 (+2)
SViT-ID	0.515 (−.187)	73.8 (+2.2)

(b) Action Recognition in SSv2.

Model	Top-1
SlowFast, R101†	63.1
ViViT-L †	65.4
MViTv1 †	64.7
MViTv2 †	68.1
MViTv2 MT	68.3
SViT-SFT	68.1 (∼0.0)
SViT-DD	68.2 (+0.1)
SViT-ID	68.9 (+0.8)

(c) Action Recognition in Diving48.

Model	Top-1
SlowFast, R101 †	77.6
TimeSformer †	74.9
TimeSformer-HR †	78.0
MViTv2 †	73.1
MViTv2 MT	75.5
SViT-SFT	77.9 (+4.8)
SViT-DD	79.8 (+6.7)

Table 2: **Results on Ego4D, SSv2, and Diving48 datasets.** Evaluation metrics for the Ego4D object state change task are absolute error in seconds (temporal loc. error. Lower is better) for temporal localization, and top-1 accuracy (Cls. top-1) for state change classification. We denote methods that do not use any additional structured annotation with †.

3.4 Compositional & Few-Shot Action Recognition

Several video datasets define an action as a combination of a verb and a noun. One challenge then is to recognize combinations that were not seen during training. This "compositional" setting was explored in the "SomethingElse" dataset [58], where verb-noun combinations in the test data do not occur in the training data. The split contains 174 classes with 54,919/54,876 videos for training/validation. We also evaluate on the few-shot compositional action recognition task in [58] (See supplementary).

Table 1 reports the results for these two tasks. All of our variants outperform all prior works and the MViTv2 MT on the *Compositional* and *Few-shot* task. From the table, we can see that using 2% of the in-domain data as auxiliary images (SViT$\times$2%), is almost comparable to using all the available in-domain images (SViT-ID). Furthermore, the results are comparable to SViT-DD (0.7 difference) when trained on auxiliary images from different domains. In this way, our approach optimizes the use of structural information across domains. Last, SViT-DD outperforms SViT-SFT when using the same data, demonstrating the value of finetuning video and image together.

3.5 Object State Change Classification and Localization

Hands and objects are key elements in human activity. Two tasks related to hand-object interaction have recently been introduced in the Ego4D [31] dataset. The first is temporal localization, which is defined as finding key frames in a video clip that indicate a change in object state. The second is object state change classification, which indicates whether an object state has changed or not.

Table 2a reports results on the temporal localization and object state change classification tasks on the Ego4D dataset. We observe that SViT-ID and SViT-DD are very close, and that both SViT-ID and SViT-DD are better than SViT-SFT. These results are consistent with those presented in Section 3.4, which indicates that our method successfully leverages structural information across domains, and that a joint image and video finetuning is beneficial.

(a) Method components

Model	Top-1
MViTv2	63.3
MViTv2 MT	63.8
MViTv2 +Object Tokens (OT)	65.0
MViTv2 +OT +FC Loss (SViT)	65.8

(b) Images Amount

% of data	Top-1
2	65.6
25	65.6
50	65.7
100	65.8

(c) HAOG Attributes

Model	Top-1
SViT -Contact	65.5
SViT -Contact -Corresp.	65.3
SViT +Geometry	65.7
SViT	65.8

Table 3: **Ablations.** We show (a) Contributions of SViT components. (b) Amount of annotated images used in training. (c) Examine different HAOG information. The experiments are performed on the SomethingElse split. For more ablations, see supplementary.

3.6 Action Recognition

Table 2b and Table 2c report the results for standard action recognition task on the SSv2 and Diving48 datasets respectively. It can be seen that in Diving48 dataset, our method improves over the MViT-v2 baseline by +6.7%, outperforming all the methods. This demonstrates that our method has some ability to generalize the learned structural information of hand-object interactions even to fine-grained "human-motion" tasks such as Diving48. We also note that Diving48 has no structured annotations available, thus SViT-ID can not be provided. On SSv2, we overall observe a smaller improvement. SViT-ID improves the baseline by 0.8% while SViT-DD improves by 0.1%. As we observed significant improvements on the more challenging SomethingElse setup, we hypothesize that the ability to generalize to unseen combinations of objects and hands exhibited by SViT is less needed on SSv2, hence leading to relatively smaller improvements.

3.7 Ablations

We perform a comprehensive ablation study on the compositional action recognition task [58] on the SomethingElse dataset to measure the contribution of the different SViT components (Table 3).

Components of the SViT model. In Table 3a, we ablate our model to show the significance of the individual components. First, we train a standard MViTv2 in a multi-task setting (MViTv2 MT), where we use videos for video action recognition, and images for HAOG prediction. There is a small improvement (+0.5) over the baseline. We then take the MViTv2 and add the object tokens and corresponding SViT loss; this leads to a more significant improvement (+1.7). Last, when adding both the consistency loss and the object tokens, we observe the largest improvement (+2.5).

The amount of in-domain data. In Table 3b, we demonstrate the effect of the amount of in-domain image data used during training. Specifically, since "SomethingElse" includes HAOG annotations for each frame, we train our method with different amount of images (2%, 25%, 50%, and 100%) from the "SomthingElse" dataset. It can be seen that our approach takes advantage of the structured information within a small amount of data and, as a result, has enhanced sample-efficiency.

HAOG attributes. We use the object tokens to predict several aspects of the HAOG. In Table 3c we consider several variations on this prediction. The model "SViT-Contact" does not predict the contact information, resulting in -0.3% drop compared to SViT. The model "SViT+Geometry" adds to SViT additional heads that predict distances between all HAOG objects (to explicitly add geometric bias). This does not affect performance. Finally, we explore what happens when we do not provide the information about the identity of bounding boxes in the training data. Namely, we treat them as four boxes, and ask the model to match those to the object tokens, via a matching losss as in [12]. This is model "SViT-Contact -Corresp." which is quite close to "SViT-Contact", indicating that SViT can perhaps be trained with weaker labels.

4 Related Work

Joint Training with Images and Videos. Vision Transformers [20] are versatile architectures designed to be flexible in terms of input sequence length. Aside from the data "'patchification"' layer, the rest of the transformer is agnostic to the domain of the input. Consequently, an exciting direction of joint training image-video has emerged in computer vision, a direction previously less natural with convolutional architectures. Specifically, this allows for training network simultaneously on image and video, with most of the network parameters shared across both domains. Recently, multi-modal models became widely popular, from shared backbones [15, 56, 60] to separate encoders for each input modality [73, 76], and even non-vision modalities [38, 39]. In our case, we focus on video

and image input, similar to [6, 27, 86]. In contrast to these works, our training approach utilizes two domains, while one of them (image) is used only during training and is supervised by a different task. Additionally, as we demonstrate here, we are not greatly impacted by domain mismatches between images and videos, as in previous works. In a related area of research, multi-task learning [14] involves developing models that predict output for multiple tasks using a common input. Our work is different since it uses two inputs from different domains, images and videos. Similarly, our work can be viewed as a form of auxiliary task, as already shown in many works [29, 37, 72]. However, it differs in that the shared object prompts are utilized to learn the shared structured representation between video and image domains, which enhances the self-attention layers with the structured representation for the main video task.

Learning from video using other modalities. Researchers have proposed different approaches to learn from videos using audio and natural language [1, 3, 19, 61, 63, 64, 67, 81]. Vision tasks like Video Captioning [19, 24, 64, 81] and Visual Quesion Answering [2, 78] require understanding of both visual and natural language. Another line of vision and language works explored object labels, supervision of textual descriptions, or the use of region crops [49, 51], in contrast to SViT, which does not require any of these components. Additionally, Since high quality captions are expensive, researchers have proposed self-supervised ways to learn from both modalities without labeled data [61, 60, 67]. In these previous works, the two modalities must be consistent and aligned (e.g., text, audio, and RGB should be consistent for same sample). However, in our work, we do not require that the annotated images be precisely aligned with our downstream video task. Last, other new works [16, 69] explored different structured representations for videos, different from what we do, which is to leverage structure from images to videos.

Structured Models. Recently structured models have been successfully applied in many computer vision applications: video relation detection [54, 70, 75], vision and language [17, 50, 51, 77], relational reasoning [8, 9, 35, 47, 40, 66, 84, 85], human-object interactions [23, 43, 83], and even image & video generation [7, 32, 42].A recent line of works [5, 25, 26, 33, 34, 41, 57, 59, 62, 74, 79, 80, 82, 88] focuses on capturing spatio-temporal interactions for video understanding. Some of these works [33, 80, 82, 88] used static image object annotations to build the representations. However, these works differ from ours. [82] proposed the Object Transformers (OT) for long form video understanding, which model a video as a set of objects by extracting region crops from SlowFast [22] based on bounding boxes from an external detector. Contrary to this work, our transformer contains both patch tokens and object tokens, whereas the OT contains only object tokens. Additionally, our model learns the object-centric representation without the need for an external detector. Another work [88], proposed a structured model for action detection that explicitly utilizes image annotations. They use a GCN [46] to reason about actors and objects interaction, on top of an I3D network. Unlike this work, we examine how image annotations can be exploited as part of a video transformer model.

Video Understanding Models. We work on several well-benchmarked down-stream video tasks such as compositional action recognition on Something-Else dataset [58] and action recognition with focus on temporal cues de-emphasizing appearance such as in Something-Something V2 dataset [30] and Diving48 [53]. Several recent video transfromer works performed well on SSv2, such as ViViT [4], MViT [21], MFormer [65], TimeSformer [11], MViTv2 [52],and Video Swin [55]. We choose to work with MViTv2, although our method can be used on top of any of these. Additionally, we also adapt our proposed method to interesting new tasks on the Ego4D dataset [31]. We believe such fine-grained video understanding tasks require richer scene and object semantic reasoning, a requirement well afforded by our proposed Hand-Object Graph.

5 Conclusion

Video understanding is a key element of human visual perception, but modeling remains a challenge for machine vision. In this work, we demonstrated the importance of learning from a scene structure from a small set of images to facilitate video learning within or outside the domain of interest. According to our empirical study, our SViT approach improves performance on three video understanding tasks. Note, that we did not prioritize making the HAOG annotations richer, i.e., it may be possible to add other physical properties to improve structure modeling. In addition, we noticed that leveraging the structure is effective even with 2% of the in-domain frames, as well as when applying it from outside the domain. We leave for future work how different physical properties can be more effectively utilized in more complex structures. We do not anticipate a specific negative impact, but, as with any Machine Learning method, we recommend to exercise caution.

References

[1] Triantafyllos Afouras, Andrew Owens, Joon Son Chung, and Andrew Zisserman. Self-supervised learning of audio-visual objects from video. In *European Conference on Computer Vision*, pages 208–224. Springer, 2020.

[2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[3] Relja Arandjelovic and Andrew Zisserman. Look, listen and learn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 609–617, 2017.

[4] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer, 2021.

[5] Anurag Arnab, Chen Sun, and Cordelia Schmid. Unified graph structured models for video understanding. In *ICCV*, 2021.

[6] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *IEEE International Conference on Computer Vision*, 2021.

[7] Amir Bar, Roei Herzig, Xiaolong Wang, Anna Rohrbach, Gal Chechik, Trevor Darrell, and A. Globerson. Compositional video synthesis with action graphs. In *ICML*, 2021.

[8] Fabien Baradel, Natalia Neverova, Christian Wolf, Julien Mille, and Greg Mori. Object level visual reasoning in videos. In *ECCV*, pages 105–121, 2018.

[9] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[10] Rodrigo Benenson, Stefan Popov, and Vittorio Ferrari. Large-scale interactive object segmentation with human annotators. In *CVPR*, 2019.

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

[12] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 213–229, 2020.

[13] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[14] Rich Caruana. Multitask learning. In *Encyclopedia of Machine Learning and Data Mining*, 1998.

[15] Lluís Castrejón, Yusuf Aytar, Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Learning aligned cross-modal representations from weakly aligned data. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2940–2949, 2016.

[16] Brian Chen, Xudong Lin, Christopher Thomas, Manling Li, Shoya Yoshida, Lovish Chum, Heng Ji, and Shih-Fu Chang. Joint multimedia event extraction from video and article. In *EMNLP*, 2021.

[17] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *ECCV*, 2020.

[18] Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. Randaugment: Practical automated data augmentation with a reduced search space. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18613–18624. Curran Associates, Inc., 2020.

[19] Chaorui Deng, Shizhe Chen, Da Chen, Yuan He, and Qi Wu. Sketch, ground, and refine: Top-down dense video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 234–243, 2021.

[20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

[21] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *ICCV*, 2021.

[22] C. Feichtenhofer, H. Fan, J. Malik, and K. He. Slowfast networks for video recognition. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6201–6210, 2019.

[23] Chen Gao, Jiarui Xu, Yuliang Zou, and Jia-Bin Huang. Drg: Dual relation graph for human-object interaction detection. *ArXiv*, abs/2008.11714, 2020.

[24] Lianli Gao, Zhao Guo, Hanwang Zhang, Xing Xu, and Heng Tao Shen. Video captioning with attention-based lstm and semantic consistency. *IEEE Transactions on Multimedia*, 19(9):2045–2055, 2017.

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

[26] Rohit Girdhar, Deva Ramanan, Abhinav Kumar Gupta, Josef Sivic, and Bryan C. Russell. Actionvlad: Learning spatio-temporal aggregation for action classification. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3165–3174, 2017.

[27] Rohit Girdhar, Mannat Singh, Nikhila Ravi, Laurens van der Maaten, Armand Joulin, and Ishan Misra. Omnivore: A Single Model for Many Visual Modalities. In *CVPR*, 2022.

[28] Chengyue Gong, Dilin Wang, Meng Li, Vikas Chandra, and Qiang Liu. Vision transformers with patch diversification. 2021.

[29] Priya Goyal, Dhruv Kumar Mahajan, Abhinav Kumar Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6390–6399, 2019.

[30] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *ICCV*, page 5, 2017.

[31] 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.

[32] Roei Herzig, Amir Bar, Huijuan Xu, Gal Chechik, Trevor Darrell, and Amir Globerson. Learning canonical representations for scene graph to image generation. In *European Conference on Computer Vision*, 2020.

[33] Roei Herzig, Elad Ben-Avraham, Karttikeya Mangalam, Amir Bar, Gal Chechik, Anna Rohrbach, Trevor Darrell, and Amir Globerson. Object-region video transformers. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[34] Roei Herzig, Elad Levi, Huijuan Xu, Hang Gao, Eli Brosh, Xiaolong Wang, Amir Globerson, and Trevor Darrell. Spatio-temporal action graph networks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[35] Roei Herzig, Moshiko Raboh, Gal Chechik, Jonathan Berant, and Amir Globerson. Mapping images to scene graphs with permutation-invariant structured prediction. In *Advances in Neural Information Processing Systems (NIPS)*, 2018.

[36] Geoffrey E. Hinton, Nitish Srivastava, A. Krizhevsky, Ilya Sutskever, and R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *ArXiv*, abs/1207.0580, 2012.

[37] Max Jaderberg, Volodymyr Mnih, Wojciech M. Czarnecki, Tom Schaul, Joel Z. Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. *ArXiv*, abs/1611.05397, 2017.

[38] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. *arXiv preprint arXiv:2107.14795*, 2021.

[39] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International Conference on Machine Learning*, pages 4651–4664. PMLR, 2021.

[40] Achiya Jerbi, Roei Herzig, Jonathan Berant, Gal Chechik, and Amir Globerson. Learning object detection from captions via textual scene attributes. *ArXiv*, abs/2009.14558, 2020.

[41] Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action genome: Actions as composition of spatio-temporal scene graphs. *arXiv preprint arXiv:1912.06992*, 2019.

[42] Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1219–1228, 2018.

[43] Keizo Kato, Yin Li, and Abhinav Gupta. Compositional learning for human object interaction. In *ECCV*, 2018.

[44] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[45] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[46] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[47] Ranjay Krishna, Ines Chami, Michael S. Bernstein, and Li Fei-Fei. Referring relationships. *ECCV*, 2018.

[48] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.

[49] Dongxu Li, Junnan Li, Hongdong Li, Juan Carlos Niebles, and Steven C. H. Hoi. Align and prompt: Video-and-language pre-training with entity prompts. *ArXiv*, abs/2112.09583, 2021.

[50] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *ArXiv*, abs/1908.03557, 2019.

[51] Xiujun Li, Xi Yin, Chunyuan Li, Xiaowei Hu, Pengchuan Zhang, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks. *ECCV 2020*, 2020.

[52] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *CVPR*, 2022.

[53] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[54] Junwei Liang, Lu Jiang, Juan Carlos Niebles, Alexander G. Hauptmann, and Li Fei-Fei. Peeking into the future: Predicting future person activities and locations in videos. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5718–5727, 2019.

[55] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *arXiv preprint arXiv:2106.13230*, 2021.

[56] Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-task vision and language representation learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10434–10443, 2020.

[57] Chih-Yao Ma, Asim Kadav, Iain Melvin, Zsolt Kira, Ghassan Al-Regib, and Hans Peter Graf. Attend and interact: Higher-order object interactions for video understanding. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6790–6800, 2018.

[58] Joanna Materzynska, Tete Xiao, Roei Herzig, Huijuan Xu, Xiaolong Wang, and Trevor Darrell. Something-else: Compositional action recognition with spatial-temporal interaction networks. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[59] E. Mavroudi, Benjamín Béjar Haro, and René Vidal. Representation learning on visual-symbolic graphs for video understanding. In *ECCV*, 2020.

[60] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9879–9889, 2020.

[61] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2630–2640, 2019.

[62] Tushar Nagarajan, Yanghao Li, Christoph Feichtenhofer, and Kristen Grauman. Ego-topo: Environment affordances from egocentric video. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 160–169, 2020.

[63] Andrew Owens, Jiajun Wu, Josh H McDermott, William T Freeman, and Antonio Torralba. Learning sight from sound: Ambient sound provides supervision for visual learning. *International Journal of Computer Vision*, 126(10):1120–1137, 2018.

[64] Yingwei Pan, Ting Yao, Houqiang Li, and Tao Mei. Video captioning with transferred semantic attributes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6504–6512, 2017.

[65] Mandela Patrick, Dylan Campbell, Yuki M. Asano, Ishan Misra Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and Joao F. Henriques. Keeping your eye on the ball: Trajectory attention in video transformers, 2021.

[66] Moshiko Raboh, Roei Herzig, Gal Chechik, Jonathan Berant, and Amir Globerson. Differentiable scene graphs. In *WACV*, 2020.

[67] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[68] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[69] Arka Sadhu, Tanmay Gupta, Mark Yatskar, Ramakant Nevatia, and Aniruddha Kembhavi. Visual semantic role labeling for video understanding. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5585–5596, 2021.

[70] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[71] Dandan Shan, Jiaqi Geng, Michelle Shu, and David Fouhey. Understanding human hands in contact at internet scale. In *CVPR*, 2020.

[72] Trevor Scott Standley, Amir Roshan Zamir, Dawn Chen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In *ICML*, 2020.

[73] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *International Conference on Learning Representations*, 2020.

[74] Chen Sun, Abhinav Shrivastava, Carl Vondrick, Kevin Murphy, Rahul Sukthankar, and Cordelia Schmid. Actor-centric relation network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 318–334, 2018.

[75] Xu Sun, Tongwei Ren, Yuan Zi, and Gangshan Wu. Video visual relation detection via multi-modal feature fusion. *Proceedings of the 27th ACM International Conference on Multimedia*, 2019.

[76] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. pages 5099–5110. Association for Computational Linguistics, 2019.

[77] Hao Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP*, 2019.

[78] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Movieqa: Understanding stories in movies through question-answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4631–4640, 2016.

[79] Bugra Tekin, Federica Bogo, and Marc Pollefeys. H+o: Unified egocentric recognition of 3d hand-object poses and interactions. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4506–4515, 2019.

[80] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *ECCV*, 2018.

[81] Xin Wang, Wenhu Chen, Jiawei Wu, Yuan-Fang Wang, and William Yang Wang. Video captioning via hierarchical reinforcement learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[82] Chao-Yuan Wu and Philipp Krähenbühl. Towards Long-Form Video Understanding. In *CVPR*, 2021.

[83] Bingjie Xu, Yongkang Wong, Junnan Li, Qi Zhao, and M. Kankanhalli. Learning to detect human-object interactions with knowledge. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2019–2028, 2019.

[84] Keyulu Xu, Jingling Li, Mozhi Zhang, Simon S. Du, Ken ichi Kawarabayashi, and Stefanie Jegelka. What can neural networks reason about? In *International Conference on Learning Representations*, 2020.

[85] Vinicius Zambaldi, David Raposo, Adam Santoro, Victor Bapst, Yujia Li, Igor Babuschkin, Karl Tuyls, David Reichert, Timothy Lillicrap, Edward Lockhart, et al. Relational deep reinforcement learning. *arXiv preprint arXiv:1806.01830*, 2018.

[86] Bowen Zhang, Jiahui Yu, Christopher Fifty, Wei Han, Andrew M. Dai, Ruoming Pang, and Fei Sha. Co-training transformer with videos and images improves action recognition. *ArXiv*, abs/2112.07175, 2021.

[87] Chuhan Zhang, Ankush Gputa, and Andrew Zisserman. Temporal query networks for fine-grained video understanding. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[88] Yubo Zhang, Pavel Tokmakov, Martial Hebert, and Cordelia Schmid. A structured model for action detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9967–9976, 2019.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [No] The code and the data are proprietary.

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See line 341-342.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] Sec Section 3.1 (the Dataset Section).
   (b) Did you mention the license of the assets? [Yes] See the Dataset Section in supplementary.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] See the Dataset Section in supplementary.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] See the Dataset Section in supplementary.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Supplementary Material for "Bringing Image Scene Structure to Video via Frame-Clip Consistency of Object Tokens"

In this supplementary file, we provide additional information about our experimental results, qualitative examples, implementation details, and datasets. Specifically, Section A provides more experiment results, Section B provides qualitative visualizations to illustrate our approach, Section C provides additional implementation details, and Section D provides additional datasets details. Finally, we have also included our code with this supplementary material.

A Additional Experiment Results

First, we discuss the pretraining and finetuning of the SViT model variants (Section A.1). Next, we present additional ablations (Section A.2) we performed in order to test the contribution of the different SViT components.

A.1 SViT Variants

As explained in Section 3.3, we use several SViT variants, see Table 4 for details. Additionally, Table 7 provides a detailed listing of the datasets we used for each SViT variant and task.

Variant	Pretrain Auxiliary Image Data	Finetune Auxiliary Image Data
SViT-ID	In-Domain + Different-Domain	In-Domain
SViT-DD	Different-Domain	Different-Domain
SViT-SFT	Different-Domain	None

Table 4: **SViT Variants**

A.2 Additional Ablations

Next, we provide additional ablations in Table 5.

Pretraining and finetuning using in-domain auxiliary images. As part of finetuning, SViT-ID makes use of in-domain auxiliary image data, whereas for pretraining, it uses a different domain and in-domain of image data (see Table 4). It is interesting to see what happens if one uses only in-domain data for both pretraining and finetuning, and no data from external datasets. Towards this end, we consider a SViT training setup where only the in-domain images are used for both pre-training and fine-tuning. We do this both for SmthElse and Ego4D. Results are shown in Tables 5a and 5b. It can be seen that results with only in-domain image data (both pretraining and finetuning) are quite close to those of using both different and in-domain data. Thus, we conclude that the performance improvement of the in-domain and different domain are similar, demonstrating that our method is capable of benefiting both from in-domain and different-domain data.

Pertraining and Finetuning with or without auxiliary images. In Table 5c, we check the effect of using the auxiliary image data during the pretraining and finetuning for the standard SViT-ID model. Namely, during the pretraining and fine tuning processes, we examine each combination of using or not using auxiliary image data. We can see that doing finetuning and pretraining with auxiliary images yields the best results (65.8). Alternatively, only pretraining with auxiliary images (64.2) or only finetuning with auxiliary images (64.4) are advantageous compared to not using any auxiliary images. In fact, not using images at all is equivalent to MViT-v2 with additional tokens, and has no benefit over MViT-v2.

Consistency Loss. In Table 5d we consider two different types of frame-clip consistencies. The first is patch token consistency, which is simply replacing the object tokens with the patch tokens in the frame-clip consistency loss and the second is object token consistency (as described in Equation 7 in the main paper). Comparing the results to not using any frame-clip consistency (65.0), we note

that object token consistency is beneficial (+0.8), but patch consistency decreases the performance (-6.3). We hypothesize that patch consistency may reduce patch diversification within the frames, thus reducing performance. A recent study [28] has demonstrated that vision transformers tend to map different patches into similar latent representations, which results in information loss and performance degradation.

The minimum requirements to make the model work. In Table 6a we aim to systematically demonstrate the minimum requirements necessary for the model to function reasonably well. We explored the minimum requirements to make our model work. We provide the following experiments: (i) A ratio of 1 image to X videos (where X is 1,10, and 100): 65.6 (1-to-1), 64.8 (1-to-10), and 61.55 (1-to-100). See Table 6a. The results indicate that using too few annotated images (1 image per 100 videos) may result in degradation, but using a relatively small number of images (1 image per 10 videos) is sufficient to achieve good improvement.

Robustness to multiple domains. In Table 6b We performed experiments on SomethingElse (ground-truth annotations available) where hands or objects were not visible in order to verify the usefulness of our approach: (i) After filtering the videos containing (annotated) hands, we tested our model and the MViTv2 model on the filtered hand split. MViTv2 achieved an accuracy of 64.6 while our model achieved 66.5. This implies an improvement of +1.9. (ii) After filtering the videos containing (annotated) objects in more than 40% of the frames, we tested our model and the MViTv2 model on the filtered object split. MViTv2 achieved an accuracy of 64.9 while our model achieved 67.0. This implies an improvement of +2.1. We can observe that our model outperforms the baseline even when there are no objects or hands in the videos, demonstrating the robustness of our approach. The total improvement (+1.9 and +2.1) is also a little bit lower than before the filtering (+2.5), which indicates that there has been a slight degradation. Even so, it is still valuable to use the hand-object graphs.

Frame-clip consistency loss. In order to validate frame-clip consistency loss, we evaluate the following experiments: (i) We perturb the image temporal position without consistency loss. We refer to this version as SViT-Perturb (and similarly SViT-Perturb-DD and SViT-Perturb-ID). (ii) The HAOG annotations are extrapolated from one random frame of a video. This serves as additional supervision (without the consistency loss) since the HAOG annotations correspond to the video frames (we note that SViT does not require such correspondence since it uses only HAOG annotations from single images). (iii) We predict the HAOG of a random frame in a video, and then duplicate it over the temporal dimension and use it in the same manner as in the consistency loss. We find that these three baselines lead to worse performance. For the first experiment, we obtained SViT-Perturb-DD with 64.1 (while SViT-DD obtained 65.1), and SViT-Perturb-ID with 65.2 (while SViT-ID with consistency loss got 65.8). For the second experiment, the proposed baseline achieved 65.0 compared to our SViT-ID, which achieved 65.8 (and does not require correspondence). Taken together, this demonstrates the importance of our frame-clip consistency loss.

SViT-2% for Ego4D, SSV2, and Diving48 datasets. In Table 6c we evaluated SViT-2% on the Ego4d, SSv, and Diving48 datasets. The SViT-2% in Diving48 shows an improvement of +5.8 compared to the MViTv2 baseline (while SViT shows an improvement of +6.7). For SSv2, the SViT-2% experiment results in an improvement of +0.6 compared to the MViTv2 baseline (while SViT shows an improvement of +0.8). For Ego4D, the SViT-2% experiment results in an improvement of +1.8 compared to the MViTv2 baseline for object state change classification and an improvement of 0.163 compared to the MViTv2 baseline for object state change temporal localization. It can be seen that the SViT-2% also appears effective and provides similar improvements on other datasets to the results shown in Table 1.

HAOG "distillation". In this experiment we first train a vision transformer to predict the HAOG task alone and then fine-tune its backbone network for the video-related task. We evaluated and achieved 62.3 compared to 63.3 of MViTv2. This indicates that training HAOGs as a form of “distillation”, as we do in SViT, is indeed important.

The object tokens representations. To verify what the object tokens learned, we can evaluate the ability of object tokens to be utilized explicitly for the auxiliary task as a simple detector of hands and objects in images. This is accomplished by predicting the detections on SomethingElse based on the learned object tokens. The learned object tokens were compared to the MViTv2 model extended with regression and detection heads. Our model achieved an mAP of 16.8, while the proposed baseline achieved a similar result with an mAP of 15.5. These results suggest that the object tokens learn meaningful and useful representations.

(a) ID pretraining on **Smthelse**.

Model	Pretraining Aux. Images	Top-1
SViT-ID	diff-domain	65.8
SViT $\times 2\%$	diff-domain	65.1
SViT-ID	in-domain	65.6
SViT $\times 2\%$	in-domain	65.1

(b) ID pretraining on **Ego4D**.

Model	Pretraining Aux. Images	Temporal Loc. Error	Cls. Top-1
SViT-ID	diff-domain	0.515	73.8
SViT-ID	in-domain	0.524	71.9

(c) Auxiliary images during pretraining and finetuning.

Model	Pretrain	Finetune	Top-1
SViT-ID	w/o Images	w/o Images	63.4
SViT-ID	w/o Images	w/ Images	64.4
SViT-ID	w/ Images	w/o Images	64.2
SViT-ID	w/ Images	w/ Images	65.8

(d) Consistency loss.

Model	Frame-Clip Consistency Type	Top-1
SViT-ID	Patches	58.7
SViT-ID	Objects	65.8
SViT-ID	None	65.0

Table 5: **Additional Experiments.** Unless otherwise noted, all experiments were performed on the "SomethingElse" dataset. In tables 5a and 5b, "diff-domain" refers to different-domain.

How important are the HAOGs. To examine how important is the information provided by the HAOG, we suggest learning HAOGs without any useful information. Thus, we run an experiment in which the HAOGs are completely random. This means that, for each image, a random HOAG is generated by sampling the boxes and their relationships uniformly. We refer to this experiment as SViT-Random-ID. The SViT-Random-ID result on SomethingElse is 50.6 compared to 63.3 in the SViT-ID baseline. This demonstrates that predicting the actual HAOG attributes is important but that not many annotations are necessary for the auxiliary task. One explanation may be that learning the shared structured representation between the domains is sample efficient and does not require many annotated images, which we view as a big advantage of our approach.

Computational cost. The cost computation of our approach is relatively small compared to MViTv2: +0.4% in Giga (10^9) FLOPs (MViTv2 with 64.5, while SViT with 64.7) and +5% in Mega (10^6) parameters (MViTv2 with 34.4, while SViT with 36.1).

Hands in Diving48. To further analyze SViT performance on Diving48 we provide an experiment in which we remove the "hand" tokens from the HAOG annotation during training, and an experiment in which we remove the "object" tokens from the HAOG annotation during training. The result for removing the "hand" tokens is a degradation of 1.7 in top-1 accuracy, and for removing the "object" tokens is 0.6. We hypothesize that the "human" boxes provide a prior for localization that helps in classifying diving actions (diving actions could be classified based on appearance and pose).

A.3 Confidence Intervals

We provide confidence intervals for the SViT-ID and MViTv2 results on the SomethingElse dataset, as seen in Table 1. We used $n = 10$ experiments and provide a 95% confidence intervals. The results for SViT-ID are 65.8 ± 0.44 and for MViTv2 are 63.3 ± 0.47. We can see that the intervals do not intersect, thus our improvement is significant.

A.4 Additional Results

We provide more results on 3rd-person video datasets, such as Kinetics-400 and AVA. Our results for Kinetics-400 with SViT showed an improvement of +1.5 over MViTv2. As for AVA, we evaluated the mAP and received an improvement of +0.7 over MViTv2.

(a) Images-Videos Ratio

Images/Videos	Top-1
1/1	65.6
1/10	64.8
1/100	61.6

(b) Robustness to multiple domains

Model	Objects	Hands	Top-1
MViTv2	✓	✗	64.6
SViT-ID	✓	✗	66.5
MViTv2	✗	✓	64.9
SViT-ID	✗	✓	67.0

(c) SViT-2%

Dataset	Top-1	
Diving48	78.9	
SSv2	68.7	
	Top-1	Loc. Error
Ego4D	73.4	0.539

Table 6: **Additional Experiments.** All experiments were performed on the "SomethingElse" dataset unless stated otherwise.

B Qualitative Visualizations

We present qualitative visualizations of the SViT-ID "object tokens" in videos for the following datasets: SomethingElse, Ego4D, and Diving48 in Figure 3. As explained in the main paper, two tokens correspond to the "right hand" and "left hand", whereas the other two tokens correspond to the objects they are interacting with. We can see that the "object tokens" are used to detect relevant objects, even on Diving48, where there is no human-object interaction. In Diving48, we observed a few interesting phenomena: (i) the diving persons are recognized as hands. This may be explain by the fact that skin has shared visual features with hands. (ii) The objects recognized in low scores indicate that the scene does not contain any objects. Overall, despite the issues we raised above, our model performed well on Diving48. This is evidence that our method is robust to multiple domains that do not necessarily fit with human-object graph structure (HAOG), and that it still benefits from these learned object tokens through the attention.

C Additional Implementation Details

Our SViT model can be used on top of the most common video transformers (MViT [21], TimeSformer [11], Mformer [65], Video Swin [55]). For our experiments, we choose the MViTv2 [52] model because it performs well empirically. These are all implemented based on the MViTv2 [52] library (available at `https://github.com/facebookresearch/mvit`), and we implement SViT based on this repository. Next, we elaborate on the additional implementation details for each dataset, including details about the dataset description, optimization, and training and inference.

C.1 Ego4D

Dataset. Ego4D [31] is a new large-scale dataset of more than 3,670 hours of video data, capturing the daily-life scenarios of more than 900 unique individuals from nine different countries around the world. The videos contain audio, 3D meshes of the environment, eye gaze, stereo and/or synchronized videos from multiple egocentric cameras.

Metrics. In the Object State Change Temporal Localization task, the absolute error (in seconds) is used for evaluation. In the Object State Change Classification task, the top-1 accuracy is used for evaluation.

Optimization details. We train using 16 frames with sample rate 4 and batch-size 128 (comprising 64 videos and 64 images) on 8 RTX 3090 GPUs. We train our network for 10 epochs with Adam optimizer [45] with a momentum of $9e{-}1$ and Gamma $1e{-}1$. Following [52], we use $lr = 1e{-}4$ with half-period cosine decay. Additionally, we used Automatic Mixed Precision, which is implemented by PyTorch.

Training details. We use a standard crop size of 224, and we jitter the scales from 256 to 320. Additionally, we set $\lambda_{Con} = 10, \lambda_{HAOG} = 5, \lambda_{Vid} = 1$.

Inference details. We follow the official evaluation, both for the state change temporal localization and the state change classification tasks, available at `https://github.com/EGO4D/hands-and-objects`.

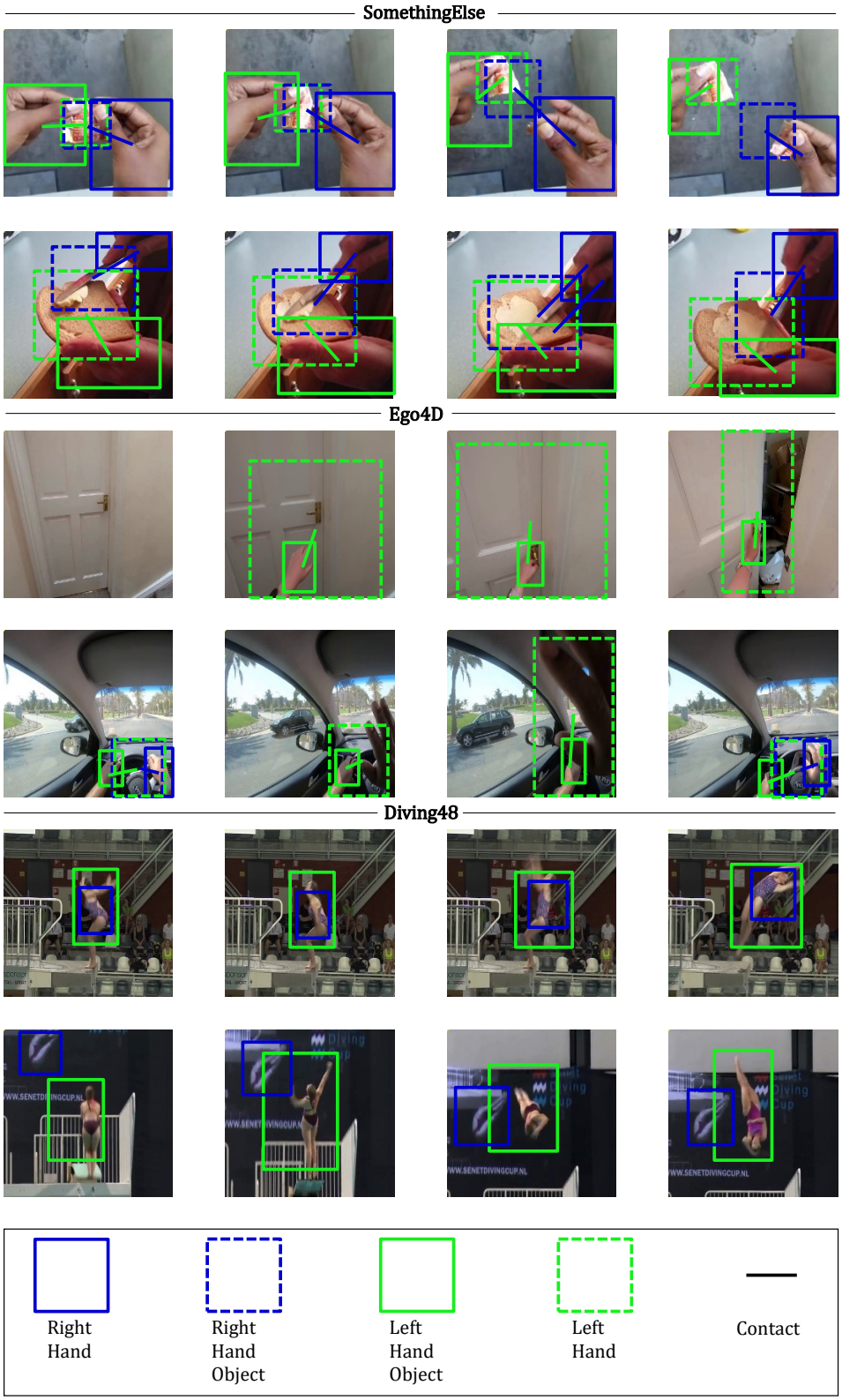

Figure 3: **Qualitative visualization of the "object tokens"**. The left object and hand prediction is visualized with **green bounding box**, the right object and hand prediction is visualized with **blue bounding box**. A line connecting between two boxes indicates a "contact" prediction between the two objects. Only boxes with a score greater than 0.6 are plotted.

C.2 Diving48

Dataset. Diving48 [53] contains 16K training and 3K testing videos spanning 48 fine-grained diving categories of diving activities. For all of these datasets, we use standard classification accuracy as our main performance metric.

Optimization details. We train using 16 frames with sample rate 4 and batch-size 128 (comprising 64 videos and 64 images) on 8 RTX 3090 GPUs. We train our network for 10 epochs with Adam optimizer [45] with a momentum of $9e-1$ and Gamma $1e-1$. Following [52], we use $lr = 1e-4$ with half-period cosine decay.

Training details. We use a standard crop size of 224 for the standard model and jitter the scales from 256 to 320. Additionally, we use RandomFlip augmentation. Finally, we sample the T frames from the start and end diving annotation time, following [87]. We set $\lambda_{Con} = 10, \lambda_{HAOG} = 5, \lambda_{Vid} = 1$.

Inference details. We take 3 spatial crops per single clip to form predictions over a single video in testing, as in [11].

C.3 SomethingElse

Dataset. The SomethingElse dataset [58] contains 174 action categories with 54,919 training and 57,876 validation samples. The proposed compositional [58] split in this dataset provides disjoint combinations of a verb (action) and noun (object) in the training and testing sets. This split defines two disjoint groups of nouns $\{\mathcal{A}, \mathcal{B}\}$ and verbs $\{1, 2\}$. Given the splits of groups, they combine the training set as $1\mathcal{A} + 2\mathcal{B}$, while the validation set is constructed by flipping the combination into $1\mathcal{B} + 2\mathcal{A}$. In this way, different combinations of verbs and nouns are divided into training or testing splits.

Few Shot Compositional Action Recognition. As mentioned in Section 3.4 of the main paper, we also evaluate on the few-shot compositional action recognition task in [58]. For this setting, we have 88 *base* action categories and 86 *novel* action categories. We train on the base categories (113K/12K for training/validation) and finetune on few-shot samples from the novel categories (for 5-shot, 430/50K for training/validation; for 10-shot, 860/44K for training/validation).

Optimization details. We train using 16 frames with sample rate 4 and batch-size 128 (comprising 64 videos and 64 images) on 8 RTX 3090 GPUs. We train our network for 100 epochs with Adam optimizer [45] with a momentum of $9e-1$ and Gamma $1e-1$. Following [52], we use $lr = 1e-4$ with half-period cosine decay. Additionally, we used Automatic Mixed Precision, which is implemented by PyTorch.

Regularization details. We use weight decay of $1e^{-4}$, and a dropout [36] of $5e-1$ before the final classification.

Training details. We use a standard crop size of 224, and we jitter the scales from 256 to 320. Additionally, we set $\lambda_{Con} = 10, \lambda_{HAOG} = 5, \lambda_{Vid} = 1$.

Inference details. We take 3 spatial crops per single clip to form predictions over a single video in testing.

C.4 Something-Something v2

Dataset. The SSv2 [58] contains 174 action categories of common human-object interactions.

Optimization details. For the standard SSv2 [58] dataset, we train using 16 frames with sample rate 4 and batch-size 128 (comprising 64 videos and 64 images) on 8 RTX 3090 GPUs. We train our network for 100 epochs with Adam optimizer [45] with a momentum of $9e-1$ and Gamma $1e-1$. Following [52], we use $lr = 1e-4$ with half-period cosine decay. Additionally, we used Automatic Mixed Precision, which is implemented by PyTorch.

Regularization details. We use weight decay of $1e-4$, and a dropout [36] of $5e-1$ before the final classification.

Training details. We use a standard crop size of 224, and we jitter the scales from 256 to 320. Additionally, we use RandAugment [18] with maximum magnitude 9. We set $\lambda_{Con} = 10, \lambda_{HAOG} = 5, \lambda_{Vid} = 1$.

Inference details. We take 3 spatial crops per single clip to form predictions over a single video in testing as done in [52].

D Additional Datasets Details

Here provide detailed information about the "auxiliary image datasets" (Section D.1) as well as the licenses and privacy policies for these datasets (Section D.2).

D.1 Auxiliary Image Datasets

Video Dataset	Task	Model	Pretrain Auxiliary Images	Finetune Auxiliary Images
SmthElse	CAR	SViT-SFT	Ego4D, 100DOH	-
SmthElse	CAR	SViT-DD	Ego4D, 100DOH	Ego4D, 100DOH
SmthElse	CAR	SViT$\times$2%	Ego4D, 100DOH	SmthElse$\times$2%
SmthElse	CAR	SViT-ID	Ego4D, 100DOH	SmthElse
Ego4D	SCTL	SViT-SFT	Ego4D, 100DOH	-
Ego4D	SCTL	SViT-DD	100DOH	100DOH
Ego4D	SCTL	SViT-ID	Ego4D, 100DOH	Ego4D
Ego4D	SCC	SViT-SFT	Ego4D, 100DOH	-
Ego4D	SCC	SViT-DD	100DOH	100DOH
Ego4D	SCC	SViT-ID	Ego4D, 100DOH	Ego4D
SSv2	AR	SViT-SFT	Ego4D, 100DOH	-
SSv2	AR	SViT-DD	Ego4D, 100DOH	Ego4D, 100DOH
SSv2	AR	SViT-ID	Ego4D, 100DOH	SSv2
Diving48	AR	SViT-SFT	Ego4D, 100DOH	-
Diving48	AR	SViT-DD	Ego4D, 100DOH	Ego4D, 100DOH

Table 7: **Auxiliary Image Datasets** The table describes which image datasets were you used for which SViT setup. CAR refers to Compositional Action Recognition, AR refers to Action Recognition, SCTL to State Change Temporal Localization, SCC to State Change Classification

In Table 7 we explicitly describe the auxiliary image datasets used in each experiment. As an example, in row 1, we describe the data used for training the SViT-SFT model for the Compositional Action Recognition (CAR) task on the SomethingElse dataset. We note that we began by pretraining on Ego4D and 100DOH.

Image/Video annotation naming conventions. The auxiliary image data we used is originated in video frames. The main difference between video frames and a batch of images is the temporal information. Since we do not use the temporal order of the annotated video frames, and practically use them as images, we refer to them in the paper as "image annotations".

D.2 Licenses and Privacy

The license, PII, and consent details of each dataset are in the respective papers. In addition, we wish to emphasize that the datasets we use do not contain any harmful or offensive content, as many other papers in the field also use them. Thus, we do not anticipate a specific negative impact, but, as with any Machine Learning method, we recommend to exercise caution.