This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Modeling Video as Stochastic Processes for Fine-Grained Video Representation Learning

Heng Zhang[1,2*] Daqing Liu[3*] Qi Zheng[4] Bing Su[1,2†]
[1] Gaoling School of Artificial Intelligence, Renmin University of China
[2]Beijing Key Laboratory of Big Data Management and Analysis Methods
[3] JD Explore Academy, JD.com [4] The University of Sydney
zhangheng@ruc.edu.cn, {liudq.ustc, subingats}@gmail.com, qi.zheng@sydney.edu.au

Abstract

A meaningful video is semantically coherent and changes smoothly. However, most existing fine-grained video representation learning methods learn frame-wise features by aligning frames across videos or exploring relevance between multiple views, neglecting the inherent dynamic process of each video. In this paper, we propose to learn video representations by modeling Video as Stochastic Processes (VSP) via a novel process-based contrastive learning framework, which aims to discriminate between video processes and simultaneously capture the temporal dynamics in the processes. Specifically, we enforce the embeddings of the frame sequence of interest to approximate a goal-oriented stochastic process, i.e., Brownian bridge, in the latent space via a process-based contrastive loss. To construct the Brownian bridge, we adapt specialized sampling strategies under different annotations for both self-supervised and weakly-supervised learning. Experimental results on four datasets show that VSP stands as a state-of-the-art method for various video understanding tasks, including phase progression, phase classification, and frame retrieval. Code is available at https://github.com/hengRUC/VSP.

1. Introduction

Fine-grained video representation learning [11] is one of the fundamental problems in computer vision, which has great practical value in various real-world applications such as action phase classification [11, 40], phase boundary detection [26], and video object segmentation [7, 9, 22, 33]. The way to model videos, especially the temporal dynamics, is the core problem of video representation learning and is highly relevant to available data annotations. Pioneer works [4,29] directly model video as 3D data where temporal is one dimension, and they require large-scale human-generated annotations for representation learning. However, it is labor-intensive and time-consuming to collect those annotations. Besides, human-generated annotations hinder domain generalization to multiple downstream tasks.

To alleviate the requirement on labeled data, some recent works [11–13] model the video alignment (Figure 1(a)) across the temporal dimensions by the cycle-consistency loss [11] or temporal alignment loss [13]. Their basic assumption is that two videos of the same action can be aligned over temporal ordering in the embedding space, and the latent correspondences across sequence pairs can be regarded as a supervisory signal. However, these methods essentially work in a weakly-supervised manner that requires video-level annotations to construct video pairs, impeding their application in the real-world scene where the semantic labels are absent.

As an alternative, self-supervised video representation learning [5, 26] explores the view relevance (Figure 1(b)) between two augmented views of one video. By modeling video as a sequence along the temporal dimensions, they elaborately construct two views through a series of spatio-temporal data augmentations. The training objective is to encourage the relevance of two augmented views to conform to their assumptions, *e.g.*, spatio-temporal contrast [26] or similarity distribution [5]. However, those methods are sensitive to complex hand-craft view augmentation thus suffering from sub-optimal performance.

As crucial and intrinsic cues, the dynamics of videos impose temporal correlations among successive frames. Therefore, the evolution process of the corresponding fine-grained representations should follow coherent constraints, which can be modeled as a stochastic process. To this end, we propose a new perspective that considers Video as Stochastic Processes (VSP) to explicitly capture the temporal dynamics of videos by exploring process agreement

*Equal contributions.
†Corresponding author.

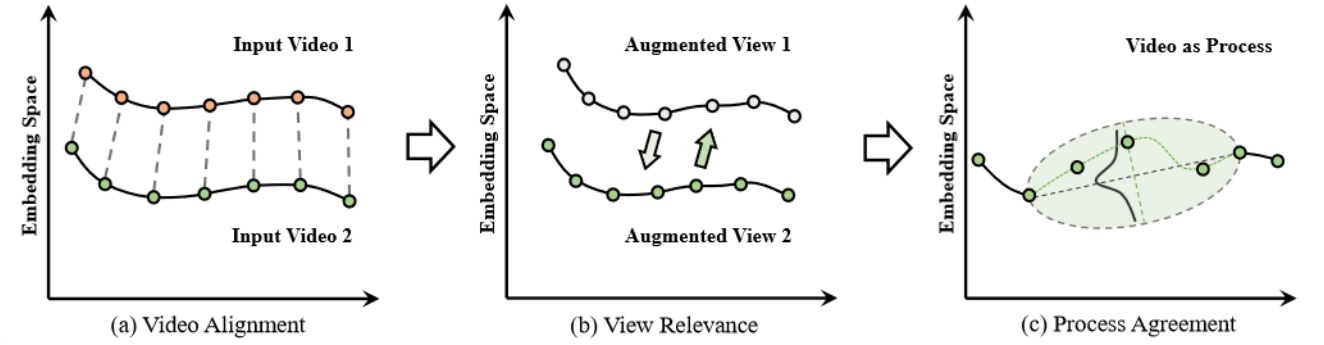


Figure 1. The evolution of fine-grained video representation learning. **(a)** *Video alignment* (*e.g.*, TCC [11], LAV [13]) enforces two videos from the same action aligned temporally. **(b)** *View relevance* (*e.g.*, TCN [26], CARL [5]) enforces the relevance of two augmented views conform to specific assumptions. **(c)** The proposed *process agreement* models video as stochastic process and enforces an arbitrary frame to agree with a time-variant Gaussian distribution conditioned on the start and end frames.

(Figure 1(c)). The basic assumption is that a video phase is coherent and smoothly changes from the start to the end, which is essentially a goal-oriented stochastic process that neighboring points are similar to each other and their coherent changes abide by the direction of the endpoint. For example, a baseball pitching video demonstrates a series of continuous movements as the ball flies out of the hand. Specifically, we model a video phase as a goal-oriented stochastic process, *i.e.*, the Brownian bridge [3, 25], where the frame representations in the latent embedding space are expected to be smooth temporal dynamics conditioned on the fixed start and end frames. With this intuitive assumption, an arbitrary frame is enforced to be like a noisy linear interpolation between the start and end frames with uncertainty in a latent space, *i.e.*, agree with a time-variant Gaussian distribution. By modeling video as stochastic processes, the proposed method captures the dynamics of each action and establishes dependencies between video frames as well as the semantic consistency of the whole video. Compared with video alignment which assumes pairing videos can be temporally aligned or view relevance which assumes two augmented views are relevant, VSP only requests process agreement that assumes the internal frames agree with the start and end frames, discarding the expensive annotated video pairs or hand-crafted view pairs.

The implementation of VSP follows a process-based contrastive learning framework where each sample is a frame triplet (start, internal, end). The start and end frames of each sample are identified as the beginning and end of the Brownian Bridge. The positive samples are the frame inside the Brownian bridge while the negatives are outside ones. The training objective is to enforce the positive samples conform to the distribution of the target Brownian bridges process while the negative samples stay away from it. Benefiting from the tunability of the start and end points of the Brownian bridge, VSP is versatile for various annotation situations. For the most generic situations where human annotations are not accessible, VSP works in a self-supervised manner by randomly sampling the triplets with an empirical length as Brownian bridges. With the phase-level annotations, VSP gains more powerful representations by taking each phase as a Brownian bridge in a weakly-supervised manner. As for the frame-level annotations, the proposed process-based contrastive objective serves as the regularization term of conventional contrastive losses.

The main contributions are summarized as follows:

- We propose a novel fine-grained video representation learning framework that models Video as Stochastic Processes (VSP) by enforcing frame sequences to conform to Brownian bridge distributions via a process-based contrastive loss.
- We adopt specialized sampling strategies for different types of annotated data by adjusting the Brownian bridge and therefore acquire favorable video representations in both self-supervised and weakly-supervised manners.
- To the best of our knowledge, we are the first to model video as a stochastic process and achieve state-of-the-art performance on various fine-grained video understanding tasks across four widely-used datasets.

2. Related Works

Weakly-supervised Learning in Videos. Previous weakly-supervised works usually leverage video-level annotations to obtain a video pair of the same action and learn temporally fine-grained representations in the alignment of the video pairs. For temporal alignment, TCC [11] designs a proxy task implemented by matching frame correspondences across the video pair with a cycle-consistency loss. Different from TCC which aligns a video pair frame by

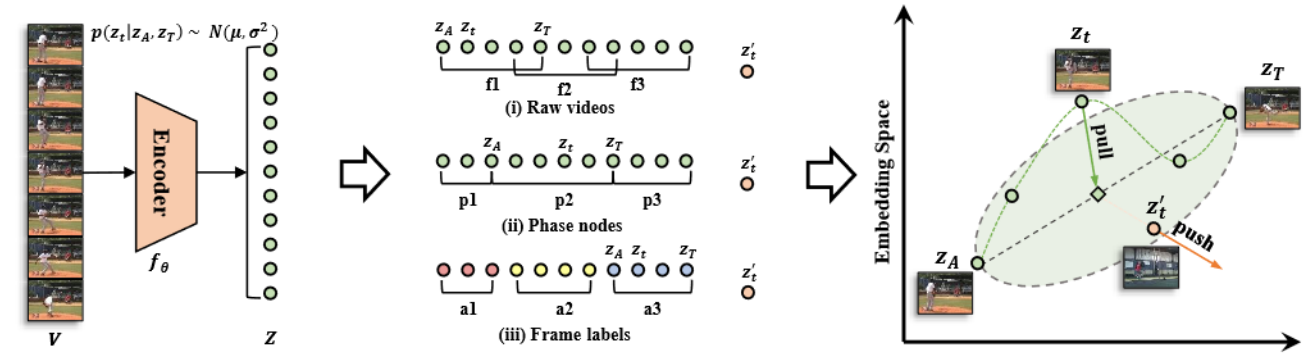


Figure 2. An overview of the proposed VSP. **(a)** We encoder frame sequences with a spatio-temporal network f_θ to get context-aware representations, which are expected to conform to the transition density of a stochastic process (Section 3.1). **(b)** Then we take special strategies for Brownian bridge construction under different annotation situations. (i) For raw videos, we build Brownian bridges by randomly sampling stated lengths and overlap thresholds. (ii) For videos with annotations on phase nodes, we build Brownian bridges by using the phase nodes as bridge boundaries. (iii) For videos with frame-level labels, we build Brownian bridges according to the frame-level annotations (Section 3.2). **(c)** Process-based Contrastive Training aims to pull the positive frames into the target Brownian bridge process while push the negatives away (Section 3.3).

frame, GTA [12] aligns the video as a whole by combining a contrastive alignment loss and a global cycle consistency loss. Similarly, LAV [13] proposes a fusion of temporal alignment loss and temporal regularization, which aims to align frame sequences and increase the similarity of temporal close frames respectively. All those methods learn representations from temporal alignment with video-level cues while our method learns from a Brownian bridge process of frame triplets without using any semantic annotations.

Self-supervised Learning in Videos. Recent self-supervised video representation learning approaches can be broadly divided into two major categories: pretext task-based approaches and contrastive learning-based approaches. 1) Spatial pretext tasks for video mainly stem from image pretext tasks, such as video colorization [31], video rotation prediction [16] and solving spatiotemporal jigsaw puzzles [1, 15, 18]. Temporal pretext tasks aim to recognize the correct or normal video, such as temporal order prediction of frame sequence [20, 21] or clips [37] from shuffled videos, playback direction prediction as a binary classification [36] and playback rate perception [6, 8, 34, 39]. 2) Most of video contrastive learning methods [10, 23, 24, 28, 38] are based on video clip discrimination where clips of the same video are positives while clips of the different videos are negatives, which is disadvantageous to temporal diversity learning of intra- and inter-clips. To alleviate this problem, a sampling-based temporal augmentation strategy was proposed in CVRL [24] to focus more on the temporally close clips. A combination of local-local and global-local temporal contrastive loss was presented in TCLR [10] to increase the temporal diversity of the learned features. In contrast, our work integrates the Brownian bridge process into contrastive learning to distinguish different action processes and maintain the temporal evaluation features in the process simultaneously.

The most recent work CARL [5] concerns the distribution of frame sequence. It compels the sequence similarity of two augmented views to conform to a prior Gaussian distribution of timestamp distance. While our approach considers the sequence distribution directly and does not rely on multiple views of a video. We relax the restrictive assumptions of the prior distribution for all intra-sequences and take each action phase as a unique Brownian bridge process wherein the distribution of frame representations changes according to timestamp.

3. Method

3.1. Video as Stochastic Processes

Given the T frames of a video as input, we encode them into latent embeddings with a frame-level video encoder [5], which is a combination of 2D DCNN [19] and Transformer [30]. Specifically, we extract per-frame features with ResNet-50 [14], then map the features into an intermediate embedding space with an MLP projection head. At last, a 3-layer Transformer concatenated with a linear layer is employed to project the encoded embeddings to the final context-aware representations, $Z = \{\boldsymbol{z}_A, \cdots, \boldsymbol{z}_T\}$.

As we discuss in the Introduction, the goal of VSP is to learn fine-grained representations from a sequence of frames that capture the dynamic evolution along the temporal dimension, which follows a stochastic process wherein frames change gradually from start to end. Based on this observation and inspired by [35], we build the stochastic

process via the Brownian bridge for videos.

Formally, taking $\boldsymbol{z}_A$, $\boldsymbol{z}_T$ as the start and end points of the Brownian bridge process respectively, the transition density of the process obeys a time-variant Gaussian distribution, given as,

$$p(\boldsymbol{z}_t|\boldsymbol{z}_A, \boldsymbol{z}_T) = \mathcal{N}\left((1-\alpha)\boldsymbol{z}_A + \alpha\boldsymbol{z}_T, \alpha(T-t)\right), \quad \text{where } \alpha = \frac{t-A}{T-A}. \tag{1}$$

Here, $\boldsymbol{z}_t$ is an arbitrary point in the process.

In terms of mean value, this density constructs $\boldsymbol{z}_t$ with a linear combination of the start and end points of the trajectory according to their relative temporal distance. $\boldsymbol{z}_t$ near the start point should be more similar to $\boldsymbol{z}_A$. In the same way, $\boldsymbol{z}_t$ near the endpoint should be more similar to $\boldsymbol{z}_T$. In terms of variance, the uncertainty of $\boldsymbol{z}_t$ conforms to a normal distribution where the value in the middle is the biggest and decreases to both ends.

3.2. Brownian Bridge Construction

In the video, we take a frame triplet as a Brownian bridge. The start and end frames of a triplet together indicate a target Brownian bridge process. As expounded in Section 3.1, the length, start, and end points of a Brownian bridge are customizable. We can build Brownian bridges unconditionally on raw videos, or conditionally on labeled videos in a more reasonable way. Here, we discuss the sampling strategies to build Brownian bridges under various annotation situations.

Raw videos. This is a more general scene where annotations are absent. We build Brownian bridges by randomly sampling frame triplets from a video with a Brownian bridge length η. For videos with less than η frames, the Brownian bridge length is the length of the video. For temporal continuity, we force adjacent Brownian bridges to have δ percent of overlap at least. This a self-supervised branch that does not rely on any semantic annotations. We denote this branch as VSP.

Phase nodes. In this setting, only the start and end points of a phase are given. we customize the Brownian bridge on phase nodes, *i.e.*, each phase is regarded as a Brownian Bridge. Thus the Brownian bridges connect via phase nodes and no overlap control is needed. Note that, we just use the start and end position information of a phase instead of the per-frame label. This is a weakly-supervised branch denoted as VSP-P.

Frame labels. Fine-grained annotations are accessible including video and phase labels, *i.e.*, frame-level annotations. We take the same strategy in VSP-P which builds the Brownian bridge on phase nodes. This branch can leverage frame labels in the process of contrastive training and we denote this branch as VSP-F.

3.3. Process Contrastive Training

After Brownian bridge construction, we next introduce the Process-based Contrastive Loss (PCL) to map frames in sequences into the latent space of the Brownian bridge and Supervise Contrastive Loss (SCL) to leverage frame annotations.

Process-based Contrastive Loss. We first define the distance between $\boldsymbol{z}_t$ and the target point in the Brownian bridge process at time t,

$$d(\boldsymbol{z}_A, \boldsymbol{z}_t, \boldsymbol{z}_T) = -\frac{1}{2\sigma^2}\left\|\boldsymbol{z}_t - (1-\alpha)\boldsymbol{z}_A - \alpha\boldsymbol{z}_T\right\|_2^2, \quad \text{where } \alpha = \frac{t-A}{T-A}. \tag{2}$$

Here, σ^2 is the variance of the Brownian bridge transition density:$\alpha(T-t)$ in Equation (1).

Next, we define the target Brownian process and its positive and negative samples in the process-based contrastive loss. A target process is denoted as $(\boldsymbol{x}_A, ..., \boldsymbol{x}_T)$ where $\boldsymbol{x}_A$, $\boldsymbol{x}_T$ are sampled from the frame sequence wherein $A<T-1$. A positive sample for the target Brownian process is a frame between $\boldsymbol{x}_A$ and $\boldsymbol{x}_T$, which is denoted as $\boldsymbol{x}_t$. Thus a positive triplet is $(\boldsymbol{x}_A, \boldsymbol{x}_t, \boldsymbol{x}_T)$. Note that, $(\boldsymbol{x}_A, \boldsymbol{x}_t, \boldsymbol{x}_T)$ indicates sampling order: $\boldsymbol{x}_t$ is sampled after $\boldsymbol{x}_A$ and $\boldsymbol{x}_T$ sampled after $\boldsymbol{x}_t$ in the video. A negative sample $\boldsymbol{x}'_t$ for the target Brownian process is a frame that does not belong to this process. A negative triplet is denoted by $(\boldsymbol{x}_A, \boldsymbol{x}'_t, \boldsymbol{x}_T)$.

Given N triplets sampled from N videos, through encoder we get their corresponding latent embeddings $\mathcal{Z} = \left\{(\boldsymbol{z}_A, \boldsymbol{z}_t, \boldsymbol{z}_T)^1, (\boldsymbol{z}_A, \boldsymbol{z}_t, \boldsymbol{z}_T)^2, ..., (\boldsymbol{z}_A, \boldsymbol{z}_t, \boldsymbol{z}_T)^N\right\}$. Specifically for a target process $(\boldsymbol{z}_A, ..., \boldsymbol{z}_T)^i$, the negative frames are provided by the other triplets, which we denote as $\mathcal{B} = \left\{\boldsymbol{z}^j_{A,t,T} | j \neq i\right\}$. Then the union of all negative frames and the positive frame for the i-th process can be denoted as $\mathcal{M} = \left\{\boldsymbol{z}^i_t\right\} \cup \mathcal{B}$. We compel the positive frame to conform to the transition density of the target Brownian bridge process described in Equation (1) while the negative frames are away from it, using the following objective:

$$\mathcal{L}^i_P = -\log \frac{\exp(d(\boldsymbol{z}^i_A, \boldsymbol{z}^i_t, \boldsymbol{z}^i_T))}{\sum_{\boldsymbol{z}'_t \in \mathcal{M}} \exp(d(\boldsymbol{z}^i_A, \boldsymbol{z}'_t, \boldsymbol{z}^i_T))}, \tag{3}$$

where $(\boldsymbol{z}^i_A, \boldsymbol{z}^i_T)$ is the start-end frame of the target Brownian bridge process $(\boldsymbol{z}_A, ..., \boldsymbol{z}_T)^i$. Thus the final loss for a training batch in VSP and VSP-P is:

$$\mathcal{L}_P = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}^i_P. \tag{4}$$

Supervised Contrastive Loss. Fine-labeled datasets provide hard positive and negative samples for contrastive loss, which is conducive to gaining powerful representations as

demonstrated in [17, 32]. To leverage those valuable cues, we design a frame-wise contrastive loss where the anchor sample is an internal frame of a phase. For the anchor frame $\boldsymbol{z}_t^i$ of the i-th triplet $(\boldsymbol{z}_A, \boldsymbol{z}_t, \boldsymbol{z}_T)^i$, the positive frames $\boldsymbol{z}_p \in \mathcal{P}$ come from the other triplets of the same action phase. While the negative frames $\boldsymbol{z}_n \in \mathcal{N}$ belong to other action phases. Then the loss of the target frame of the i-th triplet can be formulated as:

$$\mathcal{L}_S^i = -\log \frac{\sum\limits_{p\in\mathcal{P}} \exp(\boldsymbol{z}_t^i \cdot \boldsymbol{z}_p/\tau)}{\sum\limits_{p\in\mathcal{P}} \exp(\boldsymbol{z}_t^i \cdot \boldsymbol{z}_p/\tau) + \sum\limits_{n\in\mathcal{N}} \exp(\boldsymbol{z}_t^i \cdot \boldsymbol{z}_n/\tau)}. \tag{5}$$

Similarly for a training batch, $\mathcal{L}_S = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_S^i$. The goal of this objective is to pull the frames of the same subaction into a cluster. Adding our process-based contrastive objective as a regularization can further align those frames temporally in intra- and inter-action phases. And the final objective for VSP-F is:

$$\mathcal{L}_F = \frac{1}{N}\sum_{i=1}^{N}(\mathcal{L}_S^i + \mathcal{L}_P^i). \tag{6}$$

4. Experiments

4.1. Experimental Settings

Datasets. We use four action recognition datasets, namely PennAction [40], Pouring [26], IKEA ASM [2], and FineGym [27]. *PennAction* videos record humans' actions of doing sports or exercise. Following TCC [11], we use 13 actions of PennAction wherein each action owns 40-134 training and 42-116 validation videos. The phase-level annotations are provided by LAV [13] and each action is composed of 2-6 phases. The video contains 18 to 663 frames. *Pouring* videos focus on human hands interacting with objects (*i.e.*, pouring liquid from one container to another). There are 70 videos for training and 14 videos for testing in this dataset. We obtain the phase labels from TCC [11] and each video has 5 phases. The video contains 186 to 797 frames. *IKEA ASM* videos show assembling furniture by different assemblers from multiple views. It contains 371 samples of furniture assemblies and 33 action classes. Following LAV [13], we use all Kallax Drawer Shelf videos which are split into 61 training and 29 validation videos. Each video has 17 phases, which is more challenging than PennAction and Pouring. *FineGym* is a more challenging fine-grained video dataset that records the gymnastic movements of professional athletes where the semantic information is dense and frames are low redundant. The training set contains 3182 videos and the testing set contains 1442 videos following prior splits [11]. *FineGym* provides two version: *FineGym99* and *FineGym288* where the number represents sub-action classes. FineGym288 is a relatively unbalanced version compared to FineGym99.

Evaluation Metrics. For each dataset, we optimize the network on the training set and then fix its parameters. According to different metrics, a linear classifier or regressor is catenated on the frozen network and trained with annotations. We list the four evaluation metrics [11] used on the validation set. *Phase Classification* is the average per-frame phase classification accuracy. *Average Precision@K* is the fine-grained frame retrieval accuracy by computing the ratio of the correct match in retrieved K frames where K = 5, 10, 15. For all metrics, a higher score implies better performance. *Phase Progression* measures how well the representations capture the phase progress temporally. *Kendall's Tau* measures the degree of order correspondence between two sequences.

Implementation Details. We adopt the same encoder network with CARL [5], we use the ResNet-50 pre-trained on the ImageNet dataset and freeze the first four residual blocks, a 3-layer Transformer of 256 in width and 8 attention heads. In the training process, only the last residual block and its following architectures are learnable. In all experiments, our model is optimized by Adam with a learning rate of 1e-4 and a weight decay of 1e-5. We apply random crops, resize, horizontal flips, Gaussian blur, and color jittering as spatial augmentations. We implement the proposed method using PyTorch and train the model on two NVIDIA RTX 3090 GPUs for 300 epochs with batch size 128. We set $\eta = 120, \delta = 20\%$ as the default Brownian bridge length and overlap ratio in the experiments.

4.2. Comparison with State-of-the-Art Methods

We compare the following methods of multiple learning manners. *a) Video Alignment:* We compare with recent weakly-supervised learning methods, namely TCC [11], LAV [13] and GTA [12]. These methods rely on video-level annotations to pair up videos with the same action. *b) View relevance:* This learning manner requires no annotations or assumptions on datasets. We compare with prior self-supervised learning methods, namely SAL [21],TCN [26] and CARL [5]. *c) Fully-Supervised Learning:* For comparison, we train a network from scratch with explicit supervision on the phase classification task by catenating a linear classifier at the end of the network.

Phase Classification and Frame Retrieval. Table 1 shows the comparison with video-alignment-based and view-relevance-based methods on PennAction, Pouring, IKEA ASM, and FineGym99/288 datasets using phase classification and frame retrieval. Our methods outperform the other methods on both evaluation metrics on all datasets. With the help of phase or frame labels, our VSP-F and VSP-P achieve the best and second-best results respectively over all tracks. Especially, our method gains significant improve-

Method	Labels	Phase Classification				Frame Retrieval (AP@5)		
		Penn.	Pour.	IKEA.	FineGym	Penn.	Pour.	IKEA.
Video Alignment:								
TCC* [11]	Video	81.35	91.53	26.46	25.18 / 20.82	76.74	87.16	19.80
TCC [11]	Video	74.39	-	-	-	-	-	-
LAV* [13]	Video	84.25	92.84	30.43	-	79.13	89.13	23.89
LAV [13]	Video	78.68	-	-	-	-	-	-
View Relevance:								
SaL [21]	None	68.15	-	22.14	21.45 / 19.58	76.04	84.05	15.15
TCN [26]	None	68.09	89.53	26.80	20.02 / 17.11	77.84	83.56	19.15
CARL [5]	None	93.07	93.73	-	41.75 / 35.23	92.28	-	-
Process Agreement:								
VSP	None	93.12	93.85	44.29	43.12 / 36.95	92.56	91.85	26.54
VSP-P	Phase	93.27	94.08	46.77	44.58 / 38.23	93.45	93.18	28.48
VSP-F	Frame	**94.24**	**94.89**	**47.52**	**45.66 / 39.48**	**94.89**	**94.26**	**30.23**

Table 1. Comparison with video-alignment-based and view-relevance-based methods for phase classification and frame retrieval on Penn-Action, Pouring, IKEA ASM, and FineGym 99 / 288. * means special models for each action. **Best** and second best results are highlighted. The proposed process agreement outperforms both video alignment and view relevance.

Method	Labels	Penn.			Pour.			IKEA.		
		0.1	0.5	1.0	0.1	0.5	1.0	0.1	0.5	1.0
Supervised Learning	Frame	69.28	81.72	84.83	73.23	90.01	92.90	23.84	32.49	34.72
Video Alignment:										
TCC [11]	Video	79.72	81.12	81.35	90.65	91.11	91.53	24.74	25.22	26.46
LAV [13]	Video	83.56	83.95	84.25	91.61	92.82	92.84	29.78	29.85	30.43
View Relevance:										
SaL [21]	None	79.94	81.11	81.79	87.63	87.58	88.81	21.68	21.72	22.14
TCN [26]	None	81.99	82.64	82.78	89.67	87.32	89.53	25.17	25.70	26.80
Process Agreement:										
VSP	None	92.24	92.48	93.12	92.87	93.36	93.85	42.52	43.76	44.29
VSP-P	Phase	92.36	92.78	93.27	93.13	93.64	94.08	43.21	44.98	46.77
VSP-F	Frame	**92.72**	**93.62**	**94.24**	**93.93**	**94.24**	**94.89**	**45.00**	**45.97**	**47.52**

Table 2. Comparsion of Phase classification under different proportion (0.1, 0.5, 1.0) of labeled data on PennAction, Pouring, and IKEA ASM. **Best** and second best results are highlighted. The proposed process agreement significantly outperforms both video alignment and view relevance based methods.

ment on the challenging IKEA ASM and FineGym datasets, *e.g.*, 4.25% improvement on FineGym288. We further evaluate the effectiveness of our learned representations for action phase classification under 10%, 50% and 100% data protocols. As Table 2 shows, VSP outperforms the other methods on all datasets and the performances can be further improved by leveraging phase-level annotations in VSP (*i.e.,* VSP-P). Notably, using only 10% labeled data, VSP achieves comparable performance to the fully-supervised method of learning with the whole labeled data. For more frame retrieval results under AP@10 and AP@15, please refer to supplementary materials.

Phase Progression and Kendall's Tau. Due to the duplicate labels in IKEA ASM, we do not evaluate these two metrics on this dataset. As shown in Table 3, VSP surpasses all methods in both metrics on PennAction and Pouring. In the self-supervised track, our methods outperform the other methods on both datasets under all evaluation metrics except Kendall's Tau on Pouring (Only 0.002 lower). In the weakly-supervised track, VSP-F and VSP-P achieve the best and second-best results respectively on all tracks.

Method	PennAction		Pouring	
	Progress	τ	Progress	τ
TCC* [11]	0.664	0.701	0.837	0.864
TCC [11]	0.591	0.641	-	-
LAV* [13]	0.661	0.805	0.805	0.856
LAV [13]	0.625	0.684	-	-
GTA [12]	0.789	0.748	-	-
SaL [21]	0.390	0.474	-	-
TCN [26]	0.383	0.542	0.804	0.852
CARL [5]	0.918	0.985	0.935	**0.992**
VSP	**0.923**	**0.986**	**0.942**	0.990

Table 3. Comparison of *Phase Progression* and *Kendall's Tau* results on PennAction and Pouring. **Best** and second best results are highlighted. Our VSP achieves state-of-the-art performance.

η	$\delta(\%)$	Classification	AP@5	Progress	τ
60	20	90.24	90.42	0.852	0.902
120	**20**	**93.12**	**92.56**	**0.923**	**0.986**
240	20	91.38	92.32	0.875	0.918
Multiple	20	88.70	89.85	0.763	0.885
Random	-	74.83	82.18	0.618	0.654
120	0	91.08	90.74	0.821	0.893
120	10	91.73	91.43	0.862	0.910
120	**20**	**93.12**	**92.56**	**0.923**	**0.986**
120	50	91.82	91.81	0.819	0.851

Table 4. Ablation studies of Brownian bridge length η and overlap ratio δ on PennAction. 'Multiple' means the length of Brownian bridge for each training batch is chosen from $\{60, 120, 240\}$. 'Random' means that the bridge length is a random integer between [1, 240].

$\mathcal{L}_S$:$\mathcal{L}_P$	Classification	AP@5	Progress	τ
1:0.1	**94.45**	93.48	0.647	0.712
1:10	91.32	91.76	0.865	0.990
1:1	94.24	**94.89**	**0.952**	**0.994**

Table 5. Ablation studies of the weights of $\mathcal{L}_S$ and $\mathcal{L}_P$ in VSP-F.

Loss	Classification	AP@5	Progress	τ
$\mathcal{L}_I$	94.52	92.13	0.814	0.868
$\mathcal{L}_S$	**94.83**	93.89	0.427	0.475
$\mathcal{L}_S, \mathcal{L}_P$	94.45	**95.06**	**0.958**	**0.996**
VSP-F	94.24	94.89	0.952	0.994

Table 6. Ablation studies of VSP-F loss or traing strategy on PennAction. The third row means training with $\mathcal{L}_S$ for the first 150 epoches then with $\mathcal{L}_P$ for another 150 epoches.

	Pre.	Fine.	Cl.	Progress	τ
Penn.	w/o Pre.		93.12	0.923	0.986
	✓		92.35	0.894	0.952
	✓	✓	**93.57**	**0.944**	**0.988**
Pour.	w/o Pre.		93.85	0.942	0.990
	✓		92.17	0.887	0.941
	✓	✓	**94.90**	**0.958**	**0.992**
IKEA.	w/o Pre.		36.09	0.425	0.486
	✓		34.84	0.395	0.421
	✓	✓	**37.24**	**0.512**	**0.577**

Table 7. Pre-training on Kinetics-400. 'Cl.', 'Pre.' and 'Fine.' represent 'Classification', 'Pre-training' and 'Fine-tuning' respectively. VSP pretraining benefits video understanding tasks.

4.3. Ablation Study

We conduct ablation studies on the PennAction dataset to show the effectiveness of our design choices in Section 3.

Sampling Length and Overlap. We present the ablation results on Brownian bridge length and overlap ratio δ in Table 4. It can be summarized that either too short or too long Brownian bridge length is not conducive to representation learning. And a variable length offers no benefits. A proper overlap ratio for adjacent Brownian bridges can profit VSP training.

Weight. In Table 5, we adjust the weight of $\mathcal{L}_S$ and $\mathcal{L}_P$ to show the effect. In VSP-F, $\mathcal{L}_S$ is designed to pull the frames of the same subaction closer in the latent space. The $\mathcal{L}_P$ further aligns frames in each action along the temporal dimension, thus mainly affects sequence performance *Progress* and τ.

Mixed Training Strategy of VSP-F. As a contrast to VSP-F, we test another two strategies to leverage frame labels. *a) Intersecting Brownian bridge:* This method directly provides more hard-positives in $\mathcal{L}_P$. Specifically, given $\mathcal{P}$ triplets from $\mathcal{P}$ phases of the same subaction category, taking $P_i = (\boldsymbol{x}_A, ..., x_\iota, ..., \boldsymbol{x}_t)$ as the target process where ι represents the proportion of timestamp. We take frames at the ι of the other process in $\mathcal{P}$ as the hard positive samples of x_ι. We denote this branch as $\mathcal{L}_I$. *b) Pre-training for VSP:* Considering the stationary of the Brownian bridge process, hard positive frames may destroy the context consistency. We propose to pre-train the network with $\mathcal{L}_S$ for the first 150 epochs. Then we train the network by $\mathcal{L}_P$ for another 150 epochs. Note that, we ensure a batch includes all kinds of sub-actions and every kind owns multiple triplets to cater to the requirement of positives in those methods.

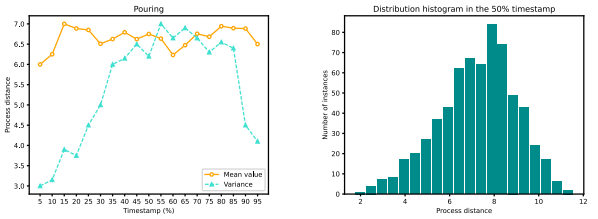
Table 6 shows the results of the two strategies and the impact of pre-training. We conclude that providing hard positives in $\mathcal{L}_I$ promotes phase classification accuracy, but at the same time reduces its performance on Phase Progression and Kendall's Tau, which demonstrates that VSP can learn typical features from hard positive frames. While embedding hard positives into the target Brownian bridge process may be detrimental to the context consistency. Pre-training with the $\mathcal{L}_S$ gathers the frames with the same action phase label into a cluster, resulting in an excellent performance in the phase classification task. After that, training with the $\mathcal{L}_P$ further aligns frames in each cluster along the temporal dimension and improves performance in Phase Progression and Kendall's Tau eventually.

4.4. Generalization Verification

To verify that our method can produce a universal model using massive raw videos, we train VSP on Kinetics-400 [4] without any semantic annotations and test on various datasets. From Table 7 we conclude that VSP can learn from large-scale unlabeled videos and the learned representations have great generalization ability. In addition, the pre-trained model achieves better performance by fine-tuning on the corresponding dataset.

4.5. Statistical Results

To verify the basic assumption that video can be taken as a goal-directed Brownian bridge process where the representations agree with a time-variant Gaussian distribution



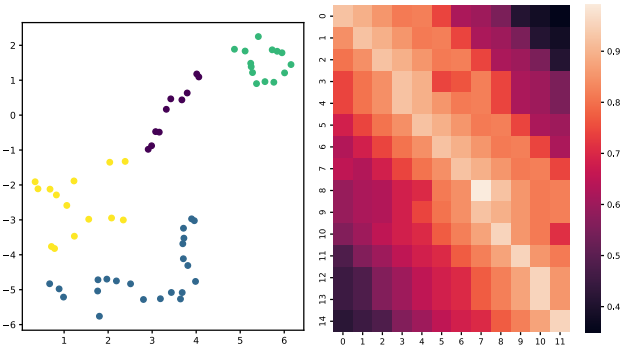
(a) Distance statistic (b) Distribution on a point

Figure 3. Statistical results (best viewed in color). (a) shows the 19 groups of mean and variance of the process distances on the whole validation set of Pouring. (b) is the frequency histogram of the process distance in the 10th group.

formulated in Equation (1), we calculate the mean and variance of the distances between frames and the target Brownian bridge process points on the whole validation set of Pouring. Specially, we take each action phase in the validation set as a Brownian bridge process, then we uniformly set 19 temporal nodes for each phase and get the corresponding 19 frames to represent the whole action phase. Next, we calculate their distances with the target points of the Brownian bridge process as described in Equation (2). Finally, we take the results of the same temporal node as a subset and calculate their mean and variance, *i.e.*, 19 groups of mean and variance. As shown in Figure 3 (a), the means of the distance in the action phases *i.e.*, approximate the linear interpolation between the start and end frames while the variances are higher in the middle of the process and lower in both ends, which is consistent with our previous hypothesis. Figure 3 (b) shows the frequency distribution histogram of the distance in the temporal node 50% timestamps, which conforms to a Gaussian distribution as described in Equation (1) and further verifies our prior assumption. More visualization results of the representations are given in the supplementary materials.

4.6. Visualization of Embeddings

We randomly select a video pair of the action *Baseball Pitch* from the PennAction dataset, which contains 4 phases and each phase represents a subaction. We first show the t-SNE visualization of embeddings for one video in Figure 4 (a). From up to down, each color represents a subaction: green for *Winding up*, purple for *Taking stride*, yellow for *Throwing*, blue for *Following through*. As the visualization result shows, the representations of frames of the same subaction gathered into a cluster. In each cluster, the representations show a consecutive motion trajectory. And between clusters, the trajectory is interconnected. These observations indicate that the learned representations are temporally consistent and distinguishable between phases in the video. We next select the second phase (*i.e.*, *Taking stride*) of the video pair and compute the similarity matrix of their



(a) t-SNE visualization (b) Similarity matrix

Figure 4. Qualitative results. (a) shows the visualization of embeddings for one video where each color indicates a subaction. (b) is the similarity matrix of the same phase in a video pair. The brighter the color, the higher the similarity.

representations. As Figure 4 (b) shows, the similarities of the video pair are closely related to timestamps: frames of the two videos with closer temporal distance are more similar and vice versa, which keeps in line with the expectations of the Brownian bridge process. This proves the temporal dynamic process and spatial consistency of the learned representations.

5. Conclusion

In this paper, we present a novel process-based contrastive learning framework named modeling Video as Stochastic Process (VSP) for fine-grained video representation learning. We capture the dynamic features by mapping the frame sequence into Brownian bridge-induced latent space where the representations change smoothly along timestamps. For this purpose, we design a process-based contrastive loss (PCL) to encourage the positive frames to fall into the target process while keeping the negatives outside, which can optionally leverage phase node cues to gain better representations. In addition, PCL can serve as the regularization term of conventional contrastive loss for the video domain to help learn temporal dynamics further. We have shown extensive experimental results on various datasets and tasks, which demonstrate the effectiveness and generalization of the representations learned by our VSP.

Acknowledgment This work was supported in part by the National Natural Science Foundation of China No. 61976206 and No. 61832017, Beijing Outstanding Young Scientist Program NO. BJJWZYJH012019100020098, Beijing Academy of Artificial Intelligence (BAAI), the Fundamental Research Funds for the Central Universities, the Research Funds of Renmin University of China 21XNLG05, and Public Computing Cloud, Renmin University of China.

References

[1] Unaiza Ahsan, Rishi Madhok, and Irfan Essa. Video jigsaw: Unsupervised learning of spatiotemporal context for video action recognition. *WACV*, 2019. 3

[2] Yizhak Ben-Shabat, Xin Yu, Fatemeh Sadat Saleh, Dylan Campbell, Cristian Rodriguez-Opazo, Hongdong Li, and Stephen Gould. The ikea asm dataset: Understanding people assembling furniture through actions, objects and pose. *WACV*, 2021. 5

[3] Rabi N. Bhattacharya and Edward C. Waymire. The brownian bridge. *Graduate Texts in Mathematics*, 2021. 2

[4] João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. *CVPR*, 2017. 1, 7

[5] Minghao Chen, Fangyun Wei, Chong Li, and Deng Cai. Frame-wise action representations for long videos via sequence contrastive learning. In *CVPR*, 2022. 1, 2, 3, 5, 6

[6] Peihao Chen, Deng Huang, Dongliang He, Xiang Long, Runhao Zeng, Shilei Wen, Mingkui Tan, and Chuang Gan. Rspnet: Relative speed perception for unsupervised video representation learning. In *AAAI*, 2021. 3

[7] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. In *NeurIPS*, 2021. 1

[8] Hyeon Cho, Taehoon Kim, Hyung Jin Chang, and Wonjun Hwang. Self-supervised spatio-temporal representation learning using variable playback speed prediction. *IEEE Access*, 2021. 3

[9] Ioana Croitoru, Simion-Vlad Bogolin, and Marius Leordeanu. Unsupervised learning from video to detect foreground objects in single images. *ICCV*, 2017. 1

[10] Ishan Rajendra Dave, Rohit Gupta, Mamshad Nayeem Rizve, and Mubarak Shah. Tclr: Temporal contrastive learning for video representation. *CVIU*, 2022. 3

[11] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. *CVPR*, 2019. 1, 2, 5, 6

[12] Isma Hadji, Konstantinos G. Derpanis, and Allan D. Jepson. Representation learning via global temporal alignment and cycle-consistency. *CVPR*, 2021. 1, 3, 5, 6

[13] Sanjay Haresh, Sateesh Kumar, Huseyin Coskun, Shahram Najam Syed, Andrey Konin, M. Zeeshan Zia, and Quoc-Huy Tran. Learning by aligning videos in time. *CVPR*, 2021. 1, 2, 3, 5, 6

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

[15] Yuqi Huo, Mingyu Ding, Haoyu Lu, Ziyuan Huang, Mingqian Tang, Zhiwu Lu, and Tao Xiang. Self-supervised video representation learning with constrained spatiotemporal jigsaw. In *IJCAI*, 2021. 3

[16] Longlong Jing, Xiaodong Yang, Jingen Liu, and Yingli Tian. Self-supervised spatiotemporal feature learning via video rotation prediction. *CVPR*, 2019. 3

[17] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *NeurIPS*, 2021. 5

[18] Dahun Kim, Donghyeon Cho, and In-So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *AAAI*, 2019. 3

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 3

[20] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Kumar Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. *ICCV*, 2017. 3

[21] Ishan Misra, C. Lawrence Zitnick, and Martial Hebert. Shuffle and learn: Unsupervised learning using temporal order verification. In *ECCV*, 2016. 3, 5, 6

[22] Seoung Wug Oh, Joon-Young Lee, N. Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. *ICCV*, 2019. 1

[23] Tian Pan, Yibing Song, Tianyu Yang, Wenhao Jiang, and Wei Liu. Videomoco: Contrastive video representation learning with temporally adversarial examples. In *CVPR*, 2021. 3

[24] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, H. Wang, Serge J. Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. *CVPR*, 2021. 3

[25] Daniel Revuz and Marc Yor. Continuous martingales and brownian motion. *Springer-Verlag, Berlin*, 1991. 2

[26] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, and Sergey Levine. Time-contrastive networks: Self-supervised learning from video. *ICRA*, 2018. 1, 2, 5, 6

[27] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Finegym: A hierarchical video dataset for fine-grained action understanding. In *CVPR*, 2020. 5

[28] Li Tao, Xueting Wang, and T. Yamasaki. Self-supervised video representation learning using inter-intra contrastive framework. *ACM MM*, 2020. 3

[29] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 1

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

[31] Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin P. Murphy. Tracking emerges by colorizing videos. In *ECCV*, 2018. 3

[32] Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. *CVPR*, 2021. 5

[33] Haochen Wang, Xiaolong Jiang, Haibing Ren, Yao Hu, and Song Bai. Swiftnet: Real-time video object segmentation. *CVPR*, 2021. 1

[34] Jiangliu Wang, Jianbo Jiao, and Yunhui Liu. Self-supervised video representation learning by pace prediction. In *ECCV*, 2020. 3

[35] Rose E. Wang, Esin Durmus, Noah D. Goodman, and Tatsunori Hashimoto. Language modeling via stochastic processes. In *ICLR*, 2022. 3

[36] D. Wei, Joseph J. Lim, Andrew Zisserman, and William T. Freeman. Learning and using the arrow of time. *CVPR*, 2018. 3

[37] D. Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. *CVPR*, 2019. 3

[38] Ceyuan Yang, Yinghao Xu, Bo Dai, and Bolei Zhou. Video representation learning with visual tempo consistency. *arXiv preprint arXiv:2006.15489*, 2020. 3

[39] Yuan Yao, Chang Liu, Dezhao Luo, Y. Zhou, and Qixiang Ye. Video playback rate perception for self-supervised spatio-temporal representation learning. *CVPR*, 2020. 3

[40] Weiyu Zhang, Menglong Zhu, and Konstantinos G. Derpanis. From actemes to action: A strongly-supervised representation for detailed action understanding. *ICCV*, 2013. 1, 5