# Fine-grained Key-Value Memory Enhanced Predictor for Video Representation Learning

Xiaojie Li
Harbin Institute of Technology,
Shenzhen
Peng Cheng Laboratory
xiaojieli0903@gmail.com

Jianlong Wu*
Harbin Institute of Technology,
Shenzhen
wujianlong@hit.edu.cn

Shaowei He
Harbin Institute of Technology,
Shenzhen
shaowei.hsw@gmail.com

Shuo Kang
Sensetime Research
21825176@zju.edu.cn

Yue Yu
Peng Cheng Laboratory
yuy@pcl.ac.cn

Liqiang Nie
Harbin Institute of Technology,
Shenzhen
nieliqiang@gmail.com

Min Zhang
Harbin Institute of Technology,
Shenzhen
zhangmin2021@hit.edu.cn

## ABSTRACT

Self-supervised learning methods have shown significant promise in acquiring robust spatiotemporal representations from unlabeled videos. In this work, we address three critical limitations in existing self-supervised video representation learning: 1) insufficient utilization of contextual information and lifelong memory, 2) lack of fine-grained visual concept alignment, and 3) neglect of the feature distribution gap between encoders. To overcome these limitations, we propose a novel memory-enhanced predictor that leverages key-value memory networks with separate memories for the online and target encoders. This design enables the effective storage and retrieval of contextual knowledge, facilitating informed predictions and enhancing overall performance. Additionally, we introduce a visual concept alignment module that ensures fine-grained alignment of shared semantic information across segments of the same video. By employing coupled dictionary learning, we effectively decouple visual concepts, enriching the semantic representation stored in the memory networks. Our proposed approach is extensively evaluated on widely recognized benchmarks for action recognition and retrieval tasks, demonstrating its superiority in learning generalized video representations with significantly improved performance compared to existing state-of-the-art self-supervised learning methods. Code is released at https://github.com/xiaojieli0903/FGKVMemPred_video.

## CCS CONCEPTS

• **Computing methodologies → Activity recognition and understanding**; • **Information systems** → *Video search.*

## KEYWORDS

Video Representation; Memory Networks; Dictionary Learning

**ACM Reference Format:**
Xiaojie Li, Jianlong Wu, Shaowei He, Shuo Kang, Yue Yu, Liqiang Nie, and Min Zhang. 2023. Fine-grained Key-Value Memory Enhanced Predictor for Video Representation Learning. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23), October 29–November 3, 2023, Ottawa, ON, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3581783.3612131

## 1 INTRODUCTION

In recent years, self-supervised learning (SSL) has gained considerable interest as a promising approach for learning spatiotemporal representations from large-scale unlabeled videos, thereby reducing the reliance on laborious manual annotation. State-of-the-art SSL frameworks, such as contrastive learning [2, 4, 5, 23, 62] and non-contrastive learning [3, 6, 16], have been successfully adapted for spatiotemporal representation learning in videos, achieving impressive performance across various downstream tasks [20, 42, 70]. However, these methods exhibit several limitations:

**(1) Insufficient Utilization of Contextual Information and Long-term Knowledge.** Videos contain valuable contextual cues, rich visual patterns, and temporal dynamics, which can enhance video representation learning and predictive capabilities. Existing SSL methods tend to neglect the significance of leveraging contextual information and relevant knowledge accumulated over the entire training process. Unlike humans who naturally integrate contextual cues and long-term memories to make informed predictions, current approaches lack mechanisms to effectively utilize this contextual knowledge in the learning process.

[*] Corresponding author.

*MM '23, October 29– November 3, 2023, Ottawa, ON, Canada.*

ACM ISBN 979-8-4007-0108-5/23/10…$15.00
https://doi.org/10.1145/3581783.3612131

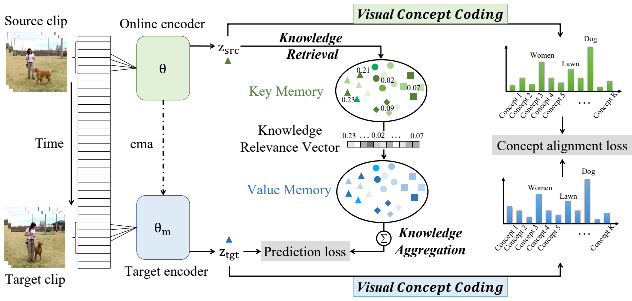


**Figure 1: Illustration of our proposed method. The fine-grained key-value memory-enhanced predictor aggregates knowledge from paired key-value memory slots, while the visual concept alignment module aligns fine-grained visual concept distributions across clips from the same video.**

**(2) Lack of Fine-Grained Visual Concept Alignment.** While existing contrastive learning methods align global high-level semantic information, they often neglect the importance of fine-grained visual concept alignment between different views of the same video. Overcoming this limitation is essential to capture intricate visual relationships and promote semantic consistency within the video representation [14].

**(3) Disregard for Feature Distribution Gap between Encoders.** Self-supervised learning methods, particularly those employing momentum encoders, aim to minimize the discrepancy between student (online encoder) and teacher (momentum-updated encoder) feature spaces. However, directly aligning features from encoders with different parameters may not effectively narrow the feature distribution gap, limiting the overall predictive capability [17, 47].

To address these limitations, we propose a novel approach that combines a key-value memory enhanced Predictor and a visual concept alignment module. The proposed predictor leverages a key-value memory network, incorporating separate key and value memories to store the learned knowledge throughout the entire training process (as depicted in Fig.1). Memory slots act as a comprehensive knowledge repository, enabling the model to access pertinent knowledge for prediction, leveraging a wide range of information from the entire dataset and facilitating knowledge transfer across different instances.

Moreover, to tackle the feature distribution gap between the online and target encoder, we utilize separate external memories. The key memory captures knowledge from the online encoder, while the value memory stores information from the momentum-updated encoder. By querying the key memory with features from the online encoder, we generate a knowledge relevance vector, which serves as a bridge to associate encoder-specific memories. The prediction features are generated by combining value memory slots using a weighted sum. Since the value memory slots are supervised by features from the target encoder, they have similar distributions to the target encoder, resulting in improved predictions with a reduced feature distribution gap.

In addition, our visual concept alignment module creates visual concept dictionaries for the online and momentum-updated encoders, encoding visual concept codes for temporally different clips. By minimizing the KL divergence between the coefficients encoded by the encoders, the alignment module promotes fine-grained semantic consistency within the video, aligning shared visual concepts between different views of the same video, such as the woman, lawn, and dog shown in Fig.1. The alignment of visual concepts also enhances the semantic richness of the knowledge stored in the memory networks, capturing intricate relationships and video-specific visual patterns.

In summary, our proposed approach makes significant advancements in self-supervised video representation learning by effectively addressing key limitations of existing methods. By leveraging contextual information, fine-grained visual concept alignment, and reducing the feature distribution gap between encoders, our method empowers video representation learning with improved predictive capabilities. Through extensive experiments on benchmark datasets, including Kinetics, UCF101, and HMDB51, we demonstrate the superiority of our approach over state-of-the-art methods in action recognition and retrieval tasks, showcasing its effectiveness in learning generalized video representations.

## 2 RELATED WORK

**Self-Supervised Learning.** Self-supervised image representation learning has witnessed remarkable progress in generating robust visual representations from unlabeled data. Among the prominent approaches, contrastive SSL methods like SimCLR [4] and MoCo [23] achieve robust feature representations by enforcing similarity between representations of the same instance from different views (positive pairs) while separating representations of different instances (negative pairs). On the other hand, non-contrastive SSL methods, including BYOL [16], DINO [3], SimSiam [6], and SwAV [2], eliminate the need for negative samples and instead focus on learning invariant features by matching positive samples, achieving comparable performance to contrastive learning methods. Momentum encoders, commonly known as slowly moving average networks, play a pivotal role in many SSL models. For instance, MoCo [23] employs momentum encoders to ensure consistent representations of a large number of negative pairs stored in the memory bank, allowing the key encoder to learn from the slowly progressing momentum encoder. Similarly, BYOL [16] predicts the output of a momentum-updated target encoder using the online encoder, while DINO [3] aligns the distribution of pseudo-classes between an online encoder and a momentum target encoder.

Compared to images, videos inherently provide richer supervision signals, such as motion changes, deformations, occlusions, and lighting variations. Various pretext tasks have been proposed to build robust spatiotemporal representations [18, 19, 29, 32, 37, 40, 65]. Contrastive SSL methods for videos have achieved remarkable success by leveraging the spatiotemporal structure of videos to generate diverse positive and negative samples [8–10, 14, 28, 35, 37, 44, 46, 49–51, 60, 68]. For example, DPC [18] extends the contrastive predictive coding framework [43] to videos by predicting dense feature representations in the future using spatiotemporal contrastive loss. MemDPC [19] introduces a memory-enhanced dense predictive coding model to handle multiple future hypotheses in the learning process. Additionally, non-contrastive SSL methods, such as $\rho$BYOL, have also been explored in the video domain, extending the BYOL framework [16] to videos by incorporating a temporally persistent objective and leveraging the temporal persistence of visual concepts. In this paper, we adopt a similar training paradigm to $\rho$BYOL, eliminating the use of negatives, and introduce key-value memory networks to enhance the MLP predictor.

**Memory Networks.** Memory networks, introduced in [55, 61], have proven to be a powerful approach for enhancing neural networks by addressing the limitations of internal memory [7, 25] in handling long-term dependencies and retaining information over time. They have demonstrated success in various tasks, including few-shot learning [30, 63], object segmentation [38], and anomaly detection [45, 69]. These networks utilize memory slots as a persistent storage mechanism that can be updated and queried, enabling the network to retain crucial information throughout the training process.

The key-value memory network [41] has proven valuable for question-answering tasks, effectively retrieving relevant memories from the key memory based on a given query and returning corresponding values from the value memory. Building on this concept, prior studies [33, 34] have also utilized key-value networks for cross-modal data, storing source modality features in the key memory and target modality features in the value memory. In this work, we adopt the key-value memory structure to capture and retain crucial information from videos, allowing our model to access pertinent knowledge from the past and enhance video representation learning.

**Dictionary Learning.** Dictionary Learning (DL) is a widely-used technique for representation learning applied in various domains, including image modeling and multi-modal representation learning [1, 11, 71]. Coupled dictionary learning has been proposed for joint representation learning in tasks involving multiple related tasks or modalities, enabling the capture of shared and specific information [26, 66]. Recent work has established connections between autoencoders (AEs) and dictionary learning, treating sparse coding and dictionary learning as neural networks [15, 54, 57]. Qian et al. [48] jointly learned visual concept dictionaries for the original video, static frame, and frame difference, with a focus on concept alignment and decoupling static and dynamic properties. In contrast, our paper adopts coupled dictionary learning to achieve fine-grained semantic alignment by aligning shared visual concepts across different views of the same video.

# 3 APPROACH

We provide an overview of existing non-contrastive self-supervised spatiotemporal representation methods (Section 3.1). Our proposed key-value memory enhanced predictor efficiently retrieves knowledge from long-term memory and addresses feature distribution gaps between the online and target encoders (Section 3.2). Additionally, the visual concept alignment module aligns shared visual concepts across video views, improving knowledge storage and prediction performance. We optimize the proposed components using an overall loss function (Fig. 2).

## 3.1 Problem Formulation

Non-contrastive spatiotemporal representation learning methods aim to train encoders that generate persistent spatiotemporal representations for different clips of the same video. The framework comprises an online encoder and a momentum-updated target encoder. The online encoder, denoted by the backbone $f_\theta$ and the projector $g_\theta$, and the target encoder, denoted by the backbone $f_{\theta_m}$ and the projector $g_{\theta_m}$, are depicted in the upper and lower parts of Fig. 2(a), respectively. The target encoder is an exponential moving average of the online encoder, updated as $\theta_m \leftarrow \lambda\theta_m + (1-\lambda)\theta$, where $\lambda \in [0, 1)$ is a momentum coefficient. Only the parameters $\theta$ are updated through back-propagation during pre-training.

For our approach, we sample two different views, the source clip $x_{\text{src}} \in \mathbb{R}^{C\times T\times H\times W}$ and the target clip $x_{\text{tgt}} \in \mathbb{R}^{C\times T\times H\times W}$, from the same video at different timestamps. After temporal-consistent augmentation to preserve motion information, the clips are fed into the online and target encoders for feature extraction. The online encoder generates the representation $y_{\text{src}} = f_\theta(x_{\text{src}}) \in \mathbb{R}^d$ and the projection $z_{\text{src}} = g_\theta(y_{\text{src}}) \in \mathbb{R}^{d_{\text{proj}}}$ for the source clip, while the target encoder produces $y_{\text{tgt}} = f_{\theta_m}(x_{\text{tgt}}) \in \mathbb{R}^d$ and $z_{\text{tgt}} = g_{\theta_m}(y_{\text{tgt}}) \in \mathbb{R}^{d_{\text{proj}}}$ for the target clip. Here, $d$ and $d_{\text{proj}}$ represent the dimensions of the feature vectors. An MLP prediction head is added on top of $g_\theta$ to transform the source clip features and align them with the target clip features. The negative cosine similarity is then minimized to achieve the alignment as follows:

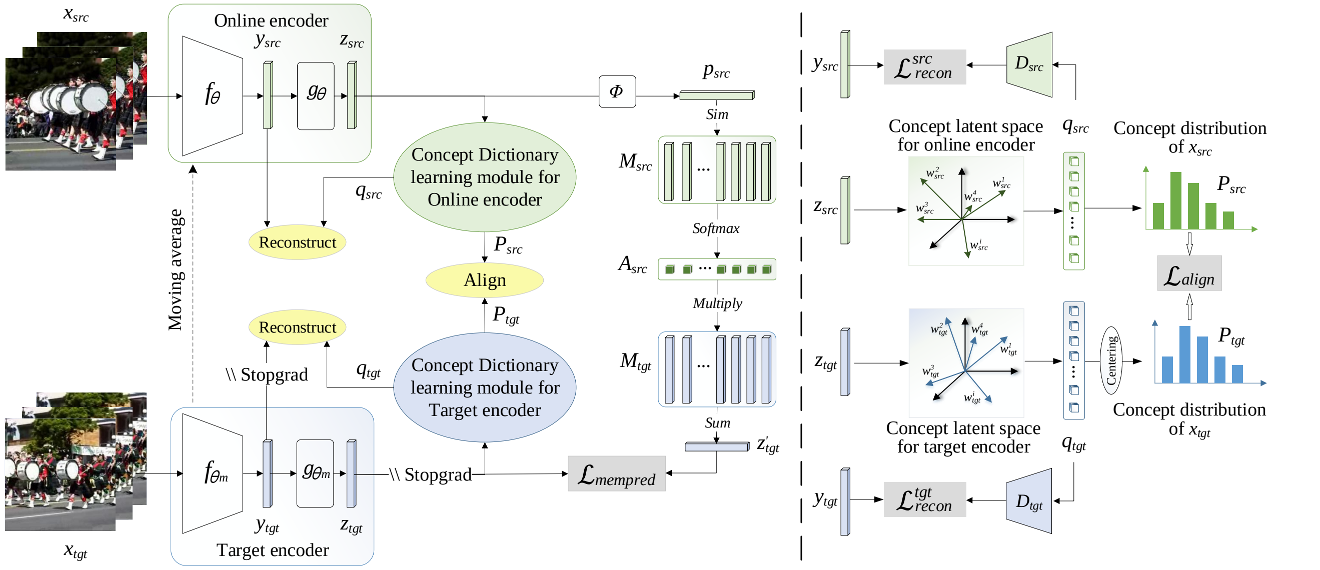
$$\mathcal{L}_{\text{pred}} = -\frac{MLP(z_{\text{src}}) \cdot z_{\text{tgt}}}{\|MLP(z_{\text{src}})\| \cdot \|z_{\text{tgt}}\|}. \tag{1}$$

In this paper, we propose a novel approach to enhance the MLP predictor utilized in existing prediction-based SSL methods for videos. We augment the MLP predictor with the ability to retrieve and integrate valuable knowledge learned throughout the entire training process. Additionally, we introduce a fine-grained visual concepts alignment mechanism to align semantic information between different clips of the same video, further improving the performance. The overall architecture of our proposed method is illustrated in Fig. 2. In Fig. 2(a), we present the pipeline of our fine-grained key-value memory enhanced predictor. In Fig. 2(b), the visual concept alignment module is illustrated. The details of our proposed method are described in the following sections.

## 3.2 Key-Value Memory Enhanced Predictor

We utilize a key-value memory-enhanced predictor to store valuable knowledge during pre-training and retrieve relevant knowledge through the key-value addressing mechanism before making predictions. Specifically, the proposed predictor consists of two encoder-specific memory networks, as depicted in Fig. 2, comprising the key memory $\boldsymbol{M}_{\text{src}} \in \mathbb{R}^{N\times d_{\text{mem}}}$ for the online encoder and the value memory $\boldsymbol{M}_{\text{tgt}} \in \mathbb{R}^{N\times d_{\text{mem}}}$ for the target encoder. Here, $N$ represents the number of memory slots, and $d_{\text{mem}}$ represents the dimension of the memory slots. Each memory network stores generic representations of its respective encoder.

During the prediction phase, the encoded feature of the source clip $z_{\text{src}}$ is used to query the entire key memory slots, yielding the knowledge relevance vector $A_{\text{src}} = [\alpha^1, \alpha^2, \ldots, \alpha^N]$. Specifically, given $z_{\text{src}}$ as the query, we first project it to match the dimension of the key memory slots $d_{\text{mem}}$ using $p_{\text{src}} = \phi(z_{\text{src}})$, where $\phi(\cdot)$ represents the projection function. Subsequently, the knowledge relevance score for the $i$-th memory slot of the key memory $\boldsymbol{M}^i_{\text{src}}$



(a) (b)

**Figure 2: The overall architecture of our proposed method. (a) Pipeline of the fine-grained key-value memory enhanced predictor. The online encoder extracts features $z_{\text{src}}$, which are used to retrieve knowledge from the key memory $M_{\text{src}}$ and generate the knowledge relevance vector $A_{\text{src}}$. The final predictions are obtained by aggregating the memory slots in the value memory $M_{\text{tgt}}$ through weighted summation based on $A_{\text{src}}$, supervised by the target feature $z_{\text{tgt}}$ using the $\mathcal{L}_{\text{mempred}}$ loss. (b) Visual concept alignment module. Two visual concept dictionaries $D_{\text{src}}$ and $D_{\text{tgt}}$ are learned to reconstruct the features $y_{\text{src}}$ and $y_{\text{tgt}}$ encoded by the backbone of each encoder, supervised by the feature reconstruction losses $\mathcal{L}^{\text{src}}_{\text{recon}}$ and $\mathcal{L}^{\text{tgt}}_{\text{recon}}$. The alignment loss $\mathcal{L}_{\text{align}}$ is employed to align the visual concept codes $q_{\text{src}}$ and $q_{\text{tgt}}$ across different views of the same video.**

is calculated as follows:

$$\alpha^i = \frac{\exp\left(r \cdot sim(\boldsymbol{M}^i_{\text{src}}, p_{\text{src}})\right)}{\sum_{j=1}^{N} \exp\left(r \cdot sim(\boldsymbol{M}^j_{\text{src}}, p_{\text{src}})\right)}, \tag{2}$$

where sim$(\cdot)$ is a cosine similarity metric and $r$ is a scaling factor.

Next, we use $A_{\text{src}}$ to address the corresponding value memory slots $\boldsymbol{M}_{\text{tgt}}$ and integrate the addressed value memory slots using a weighted sum function to get a predicted target representation $z'_{\text{tgt}}$ as follows,

$$z'_{\text{tgt}} = A_{\text{src}} \cdot \boldsymbol{M}_{\text{tgt}}. \tag{3}$$

To train the proposed predictor, we minimize the negative cosine similarity between the predicted target representation and the target representation ztgt as shown as follows:

$$\mathcal{L}_{\text{mempred}} = -\frac{z'_{\text{tgt}} \cdot z_{\text{tgt}}}{\|z'_{\text{tgt}}\| \cdot \|z_{\text{tgt}}\|}. \tag{4}$$

By minimizing the prediction loss, the learnable parameters of the key-value memory networks are updated to store representative features. The knowledge relevance addressing mechanism ensures that each pair of key-value memory slots $\boldsymbol{M}\text{src}^i$ and $\boldsymbol{M}\text{tgt}^i$ captures knowledge with the same semantic information.

### 3.3 Fine-Grained Visual Concept Alignment

We introduce the visual concept alignment module as a key component of our proposed method, which is based on coupled dictionary learning and aims to align shared visual concepts across different views of the same video, thereby enhancing the quality of knowledge stored in memory slots for improved prediction. To achieve this, we employs two fully connected layers, $W_{\text{src}}$ for the source encoder and $W_{\text{tgt}}$ for the target encoder, to generate latent concept codes for the source and target clips, respectively. The calculations are defined as:

$$q^k_{\text{src}} = w^k_{\text{src}} \cdot z_{\text{src}},\ q^k_{\text{tgt}} = w^k_{\text{tgt}} \cdot z_{\text{tgt}}, \tag{5}$$

where $w^k_*$ denotes the $k$-th column of the fully connected layer. Each code generator has a dimension of $d_{\text{dict}}$, and we set a large number for the number of visual concepts, such as 4096, to enable an enhanced representation of the data and capture low-dimensional structures effectively.

We supervise the training of the code generators by reconstructing the original features using visual concept codes passed through dictionaries. Specifically, the generated codes, $q_{\text{src}}$ and $q_{\text{tgt}}$, passed through two distinct dictionaries, $\boldsymbol{D}_{\text{src}}$ and $\boldsymbol{D}_{\text{tgt}}$, to reconstruct the encoded features $y_{\text{src}}$ and $y_{\text{tgt}}$ obtained from the backbones. We use a two-layer MLP structure for the dictionaries. Subsequently, we utilize the $\mathcal{L}_2$ loss for optimization and apply a stop gradient to the original features, as illustrated in Eq. 6. By minimizing the reconstruction loss between the original features and the reconstructed features obtained from the visual concept codes, the concept prototypes $[w^1_{\text{src}}, w^2_{\text{src}}, \ldots, w^K_{\text{src}}]$ and $[w^1_{\text{tgt}}, w^2_{\text{tgt}}, \ldots, w^K_{\text{tgt}}]$, shown in Fig. 2, captures important information within the video.

$$\begin{aligned}\mathcal{L}_{\text{recon}} &= \mathcal{L}^{\text{src}}_{\text{recon}} + \mathcal{L}^{\text{tgt}}_{\text{recon}} \\ &= \|\boldsymbol{D}_{\text{src}}(q_{\text{src}}) - y_{\text{src}}\|^2_2 + \left\|\boldsymbol{D}_{\text{tgt}}(q_{\text{tgt}}) - y_{\text{tgt}}\right\|^2_2.\end{aligned} \tag{6}$$

To align the visual concepts across different video clips that share the same visual attributes, we adopt a knowledge distillation paradigm proposed in [3] by treating the online encoder as the student network and the momentum-updated target encoder as the teacher network. The concept codes generated by the teacher network are used to guide the training of the online encoder. To prevent collapse during aligning, we apply centering and sharpening techniques to the momentum teacher outputs, as proposed in [3]. This involves introducing a bias term $c$ and adding it to the concept codes generated by the momentum teacher encoder. The modified concept codes, denoted as $\hat{q}_{\text{tgt}}$, are computed as $\hat{q}_{\text{tgt}} = q_{\text{tgt}} + c$. The bias term $c$ is updated using an exponential moving average function with a parameter $m \in [0, 1)$ and batch size $B$. The update equation for $c$ is given by $c \leftarrow mc + (1 - m)\frac{1}{B}\sum_{i=1}^{B} q_{\text{tgt}}$. We set $m = 0.9$ by default.

Next, we compute concept distributions over $K$ dimensions, represented by $P_s$ and $P_t$, using the softmax function. These distributions are controlled by temperature parameters $\tau_s > 0$ and $\tau_t > 0$, which control the sharpness of the output distribution and enable the inputs to be described by a small number of dominant concepts. These distributions capture the importance of each concept in representing the visual features of the source and target clips, respectively. By default, we set $\tau_s = \tau_t = 0.05$. Specifically, for $k$-th concept feature, we calculate $P_s^k$ and $P_t^k$ as follows:

$$P_s^k = \frac{\exp(q_{\text{src}}^k/\tau_s)}{\sum_{i=1}^{K}\exp(q_{\text{src}}^i/\tau_s)}, \; P_t^k = \frac{\exp(\hat{q}_{\text{tgt}}^k/\tau_t)}{\sum_{i=1}^{K}\exp(\hat{q}_{\text{tgt}}^i/\tau_t)}. \tag{7}$$

Finally, we calculate the cross-entropy loss between the visual concept distributions to train the online encoder. The loss is defined as:

$$\mathcal{L}_{\text{align}} = -\sum_{k=1}^{K} \text{stopgrad}(P_t^k)\log P_s^k, \tag{8}$$

where the stop-gradient operator is applied to the target encoder's concept distributions to prevent gradient propagation through the target encoder.

By minimizing $\mathcal{L}_{\text{align}}$, we align the concept distributions of temporally different segments within the same video, and jointly optimize the feature representations and concept descriptions across a large set of video samples. Additionally, as depicted in Fig. 2(b), we employ encoder-specific code generators and concept dictionaries for the online and target encoders. This approach allows for simultaneous learning of the encoders' representations and concept descriptions, eliminating the need to address feature distribution gaps between the encoders. We illustrate the feature distribution gaps between encoders in Appendix A.

We combine both the reconstruction and the alignment loss functions for fine-graied visual concept alignment:

$$\mathcal{L}_{\text{concept}} = \alpha\mathcal{L}_{\text{recon}} + \beta\mathcal{L}_{\text{align}}, \tag{9}$$

where $\alpha$ and $\beta$ are weighting factors that balance the contributions of the global attention loss $\mathcal{L}_{\text{recon}}$ and the local attention loss $\mathcal{L}_{\text{align}}$. We also analyze the connection and difference between memories and dictionaries in Appendix B

## 3.4 Overall Objective

By replacing the original MLP predictor with our proposed key-value memory-enhanced predictor and utilizing the visual concept alignment module to align fine-grained semantic information across temporally different clips from the same video, the overall training objective of our fine-grained key-value memory-enhanced predictor is given by:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{mempred}} + \mathcal{L}_{\text{concept}}. \tag{10}$$

# 4 EXPERIMENTS

## 4.1 Experimental Settings

**Datasets.** We evaluate our methodology on three widely-used video datasets: UCF101 [53], HMDB51 [36], and Kinetics-400 [31] (K400). UCF101 contains 1.3k videos with 101 action categories, HMDB51 includes 7k videos covering 51 categories, and K400 is a large-scale dataset with over 24k videos spanning 400 categories. For pre-training, we use either the UCF101 or K400 training set and evaluate on split 1 of UCF101 and HMDB51.

During data preprocessing, we randomly sample two clips of size $T \times s$ frames from each video. Each input clip consists of $T$ frames sampled from the raw clip with a stride of $s$. For pre-training, we apply a cropping procedure described in [14, 56] to augment the data, which involves random adjustments of the input area's scale and aspect ratio. For downstream tasks, we resize the shorter spatial side of the video within the range [256, 320] pixels for a random crop of size $224^2$, or within the range [128, 160] pixels for a random crop of size $112^2$.

**Architecture.** The encoder in our model comprises spatiotemporal convolutional neural networks with three different backbones: 3D-ResNet-18 [21, 22] (R3D-18, 31.8M parameters), R(2+1)D-18 [59] (R(2+1)D, 14.4M parameters), and Slow-R18 [13, 14] (20.2M parameters), which uses an R-18, 8×8 Slow pathway. To improve the temporal resolution of features, we modify R3D-18 by setting the temporal stride of the conv4 layer to 1 and the dilation factor to 2, following TCLR [8]. For the projection head and the baseline MLP prediction head, we employ batch normalization [27] and ReLU activations, with a two-layer MLP structure in the hidden layer, having a dimension of 4096. The default experiments utilize $d = d_{\text{proj}} = d_{\text{mem}} = d_{\text{dict}} = 256$. The scaling factor $r$ in Equ. 2 is set to 64. in The projection function $\phi$ used in our memory-enhanced predictor is a two-layer MLP with a hidden layer dimension of 256. Synchronized batch normalization is employed in the backbone, as described in [4, 14].

**Self-Supervised Learning.** We conduct training for 100 epochs on the Kinetics-400 dataset, employing a total mini-batch size of 256 and utilizing 8 Tesla V100 GPUs. For the UCF101 dataset, we perform training for 400 epochs with a total mini-batch size of 64, using 8 GeForce GTX 1080 Ti GPUs. We use a half-period cosine schedule with a base learning rate of $\eta = 1.2$ for UCF101 and $\eta = 2.4$ for Kinetics-400. LARS [67] is employed, except for bias and BN parameters, with an SGD weight decay of $10^{-4}$. We utilize a cosine schedule from 0.996 to 1 for the momentum coefficient $\lambda$ during pre-training. The balancing hyperparameters are set to $\alpha = 0.01$ and $\beta = 0.01$ by default.

**Action Recognition.** Action recognition serves as a downstream task to evaluate the effectiveness of our method. We initialize our models using pre-trained parameters, excluding the last fully-connected layer. We employ two established evaluation protocols:

(1) *Fine-tuning the entire network:* We fine-tune the entire network using action labels. The training process consists of 200 epochs with a batch size of 64. A new fully connected layer is added at the end of the pre-trained backbone for classification. Before this newly attached layer, a dropout probability of 0.5 is applied. The initial learning rate is set to 0.2, following a cosine annealing scheduler without warmup. We utilize the SGD optimizer with a momentum of 0.9 and no weight decay.
(2) *Linear probe*: We freeze the backbone and solely train the last linear classifier. Training is conducted for 100 epochs without using dropout. The remaining training strategies align with those in the fine-tuning protocol.

During inference, we follow the evaluation protocol used in [13, 14]. Specifically, we sample 10 temporal clips with 3 different spatial crops to cover the entire video. The predictions from these 30 clips are averaged, and the resulting Top-1 accuracies on the UCF101 and HMDB51 datasets are reported.

**Video Retrieval.** To evaluate the quality of the spatiotemporal features, we utilize the extracted pre-trained encoder representations without additional training. Following prior work [39, 64], we perform k-nearest neighbors (k-NN) search in the training set using video clips from the test set as queries. A global representation is obtained by averaging 10 uniformly sampled clips. A hit is counted if the category of the testing clip is present in the k-nearest neighbors. We report the Top-k recall R@k for evaluation purposes.

## 4.2 Comparison with State-of-the-art Methods

In this section, we evaluate the performance of our proposed method on action recognition and video retrieval tasks and compare it with state-of-the-art approaches. We use different input sizes for each backbone: $8 \times 224^2$ with a temporal sampling stride of 8 for Slow-R18, following the settings in [14], and $16 \times 112^2$ with a temporal sampling stride of 4 for R3D-18 and R(2+1)D.

**Action Recognition.** Tab. 1 presents the action recognition results for our method and several comparable works on the UCF101 and HMDB51 datasets, reporting Top-1 accuracy in both the *linear probe* and *fine-tune* settings. The table also provides information about the network architecture, pre-training dataset, input sizes, and evaluation protocol used in each method.
*Linear probe setting*: Our method outperforms the other methods across different backbones, achieving the highest Top-1 accuracy on both UCF101 and HMDB51. For instance, with the Slow-R18 backbone, our method achieves a Top-1 accuracy of 79.59% on UCF101 and 47.89% on HMDB51, surpassing other methods in this setting.
*Fine-tune setting*: Our method consistently achieves best performance when pre-trained on the K400 dataset. For instance, with the R(2+1)D backbone, our method achieves a Top-1 accuracy of 89.00% on UCF101 and 61.12% on HMDB51, which are the highest scores among all methods. When pre-trained on the UCF101 dataset, our method still outperforms the other methods, achieving a Top-1 accuracy of 84.32% on UCF101 and 54.21% on HMDB51 with the R3D-18 backbone. The results demonstrate the effectiveness of our method in both the *linear probe* and *fine-tune* settings across different backbone architectures and pre-training datasets.

**Table 1: Action recognition performance on UCF101 and HMDB51. Finetune ✓ indicates that the entire networks are fine-tuned end-to-end, while ✗ indicates that the backbone network is fixed and only the linear classifier is updated.**

| pre-train Dataset: K400, Finetune: ✗ | | | | |
|---|---|---|---|---|
| **Method** | **Backbone** | **Input size** | **UCF101** | **HMDB51** |
| MemDPC [19] | R3D-34 | $40 \times 224^2$ | 54.1 | 30.5 |
| CoCLR [20] | S3D | $32 \times 128^2$ | 74.5 | 46.1 |
| FAME [9] | R(2+1)D | $16 \times 112^2$ | 72.2 | 42.2 |
| DCLR [10] | R(2+1)D | $16 \times 112^2$ | 72.3 | 46.4 |
| VCL [48] | S3D | $16 \times 128^2$ | 75.1 | 47.4 |
| **Ours** | R(2+1)D | $16 \times 112^2$ | 78.19 | 47.57 |
| **Ours** | R3D-18 | $16 \times 112^2$ | 79.46 | 46.05 |
| **Ours** | Slow-R18 | $8 \times 224^2$ | **79.59** | **47.89** |

| pre-train Dataset: UCF101, Finetune: ✓ | | | | |
|---|---|---|---|---|
| **Method** | **Backbone** | **Input size** | **UCF101** | **HMDB51** |
| CoCLR [20] | S3D | $32 \times 128^2$ | 81.4 | 52.1 |
| VCL [48] | R(2+1)D | $16 \times 112^2$ | 82.1 | 49.7 |
| TCLR [8] | R(2+1)D | $16 \times 112^2$ | 82.8 | 53.6 |
| DCLR [10] | R(2+1)D | $16 \times 112^2$ | 82.3 | 50.1 |
| **Ours** | R(2+1)D | $16 \times 112^2$ | **84.32** | 53.03 |
| **Ours** | R3D-18 | $16 \times 112^2$ | 84.14 | **54.21** |

| pre-train Dataset: K400, Finetune: ✓ | | | | |
|---|---|---|---|---|
| **Method** | **Backbone** | **Input size** | **UCF101** | **HMDB51** |
| MemDPC [19] | R3D-34 | $40 \times 224^2$ | 78.1 | 41.2 |
| CoCLR [20] | S3D | $32 \times 128^2$ | 87.9 | 54.6 |
| VideoMoCo [44] | R(2+1)D | $32 \times 112^2$ | 78.7 | 49.2 |
| $\rho$MOCO [14] | Slow-R18 | $8 \times 224^2$ | 87.1 | - |
| DCLR [10] | R(2+1)D | $16 \times 112^2$ | 83.3 | 52.7 |
| TCLR [8] | R(2+1)D | $16 \times 112^2$ | 84.3 | 54.2 |
| FAME [9] | R(2+1)D | $16 \times 112^2$ | 84.8 | 53.5 |
| VCL [48] | R(2+1)D | $16 \times 112^2$ | 86.1 | 54.8 |
| **Ours** | R(2+1)D | $16 \times 112^2$ | **89.00** | **61.12** |
| **Ours** | R3D-18 | $16 \times 112^2$ | 88.29 | 57.43 |
| **Ours** | Slow-R18 | $8 \times 224^2$ | 87.50 | 57.11 |

**Table 2: Results on UCF101 for video retrieval task.**

| **Method** | **Backbone** | **R@1** | **R@5** | **R@10** | **R@20** |
|---|---|---|---|---|---|
| MemDPC [19] | S3D | 20.2 | 40.4 | 52.4 | 64.7 |
| CoCLR [20] | S3D | 53.3 | 69.4 | 76.6 | 82.0 |
| TransRank [12] | R3D-18 | 53.3 | 69.4 | 76.6 | 82.0 |
| VCLR [35] | R2D-50 | 46.8 | 61.8 | 70.4 | 79.0 |
| DCLR [10] | R(2+1)D | 54.8 | 68.3 | 75.9 | 82.8 |
| VCL [48] | R(2+1)D | 55.6 | 70.1 | 77.4 | 83.1 |
| TCLR [8] | R3D-18 | **56.2** | 72.2 | 79.0 | 85.3 |
| **Ours** | R3D-18 | 53.77 | **72.98** | **80.39** | **87.23** |
| **Ours** | R(2+1)D | 53.00 | 72.06 | 79.12 | 86.07 |

**Video Retrieval.** We present the video retrieval performance of our method on the UCF101 and HMDB51 datasets, with Recall@k (R@k) for k = 1, 5, 10, and 20 reported in Tables 2 and 3. When compared with other state-of-the-art methods such as TCLR [8], which employs short and long clips to attend to fine-grained temporal features, VideoMoCo, which improves MoCo's temporal feature representations by introducing a generator to drop out frames and using temporal decay to model key attenuation in the memory queue, VCL [48] and DCLR [10], which utilize complementary information between RGB and frame difference (FD), our proposed method consistently outperforms these methods on all datasets, particularly for R@5, R@10, and R@20.

**Table 3: Results on HMDB51 for video retrieval task.**

| Method | Backbone | R@1 | R@5 | R@10 | R@20 |
|---|---|---|---|---|---|
| MemDPC [19] | S3D | 7.7 | 25.7 | 40.6 | 57.7 |
| CoCLR [20] | S3D | 23.3 | 43.2 | 53.5 | 65.5 |
| TransRank [12] | R3D-18 | 23.3 | 43.2 | 53.5 | 65.5 |
| VCLR [35] | R2D-50 | 17.6 | 38.6 | 51.1 | 67.6 |
| DCLR [10] | R(2+1)D | 24.1 | 44.5 | 53.7 | 64.5 |
| VCL [48] | R(2+1)D | **24.4** | 45.1 | 54.5 | 66.4 |
| TCLR [8] | R3D-18 | 22.8 | 45.4 | 57.8 | 73.1 |
| **Ours** | R3D-18 | 23.62 | **50.13** | **61.32** | **73.95** |
| **Ours** | R(2+1)D | 20.66 | 46.78 | 59.61 | 73.68 |

Specifically, on the UCF101 dataset, our method using R3D-18 as the backbone achieves R@5, R@10, and R@20 values of 72.98%, 80.39%, and 87.23%, respectively, outperforming all other methods. Similarly, on the HMDB51 dataset, our method with the R3D-18 backbone achieves R@5, R@10, and R@20 values of 50.13%, 61.32%, and 73.95%, respectively, surpassing other methods as well. However, our method does not always achieve the highest R@1, suggesting there might still be room for improvement in the model’s ability to precisely retrieve the most relevant video.

## 4.3 Ablation Study

To evaluate different designs of our framework and validate the influence of different hyperparameters, we conducted an ablation study using the R3D-18 or R(2+1)D backbone pre-trained on the UCF101 dataset. We trained all models for 400 epochs with an input resolution of $16 \times 112^2$.

**Influence of Individual Components.** In this ablation study, we compare three predictor structures: MLP predictor ($\mathcal{L}_{\text{pred}}$), key-value memory enhanced predictor (KVMemPred) ($\mathcal{L}_{\text{mempred}}$), and fine-grained key-value memory enhanced predictor (FGKVMemPred) ($\mathcal{L}_{\text{mempred}} + \mathcal{L}_{\text{concept}}$). The models are evaluated using both the *fine-tune* and *linear probe* evaluation protocols.

Results in Tab. 4 show that the key-value memory enhanced predictor consistently outperforms the baseline MLP predictor, showcasing the benefits of incorporating key-value memory mechanisms. Furthermore, FGKVMemPred achieves superior performance in most cases, underscoring the advantages of our proposed method. Specifically, on the UCF101 dataset, the MLP predictor achieves a baseline fine-tuning accuracy of 82.13% with R(2+1)D and 82.53% with R3D-18. In contrast, our KVMemPred significantly improves the accuracy to 83.61% with R(2+1)D and 83.35% with R3D-18. By further aligning fine-grained semantic concepts using coupled dictionary learning, our FGKVMemPred achieves an accuracy of 84.32% with R(2+1)D and 84.14% with R3D-18. We present ablation studies about *alpha* and *beta* in Eq. 9 in Appendix C.

**Influence of Hyper-parameters.** Fig. 3 investigates the influence of various hyperparameters on our method’s performance, focusing on the number of memory slots ($N$), the size of the visual concept dictionaries ($K$), and the temperature ($\tau_s$ and $\tau_t$). All experiments are conducted using the R3D-18 backbone. The key observations from the table are:

**Table 4: Comparisons of three predictor structures, including the MLP, KVMemPred, and the FGKVMemPred predictor supervised by different loss functions.**

| Supervision | Backbone | Finetune | UCF101 | HMDB51 |
|---|---|---|---|---|
| $\mathcal{L}_{\text{pred}}$ | R(2+1)D | ✗ | 55.11 | 21.71 |
| $\mathcal{L}_{\text{mempred}}$ | R(2+1)D | ✗ | 63.67 | 34.14 |
| $\mathcal{L}_{\text{mempred}} + \mathcal{L}_{\text{concept}}$ | R(2+1)D | ✗ | **66.19** | **36.32** |
| $\mathcal{L}_{\text{pred}}$ | R3D-18 | ✗ | 58.60 | 29.28 |
| $\mathcal{L}_{\text{mempred}}$ | R3D-18 | ✗ | 67.04 | 33.75 |
| $\mathcal{L}_{\text{mempred}} + \mathcal{L}_{\text{concept}}$ | R3D-18 | ✗ | **68.89** | **36.45** |
| $\mathcal{L}_{\text{pred}}$ | R(2+1)D | ✓ | 82.13 | 52.50 |
| $\mathcal{L}_{\text{mempred}}$ | R(2+1)D | ✓ | 83.61 | **54.34** |
| $\mathcal{L}_{\text{mempred}} + \mathcal{L}_{\text{concept}}$ | R(2+1)D | ✓ | **84.32** | 53.03 |
| $\mathcal{L}_{\text{pred}}$ | R3D-18 | ✓ | 82.53 | 53.68 |
| $\mathcal{L}_{\text{mempred}}$ | R3D-18 | ✓ | 83.35 | **55.00** |
| $\mathcal{L}_{\text{mempred}} + \mathcal{L}_{\text{concept}}$ | R3D-18 | ✓ | **84.14** | 54.21 |

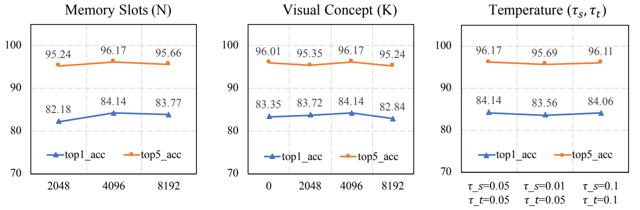


**Figure 3: Effect of hyperparameters $N$, $K$, $\tau_s$, and $\tau_t$. The values for each hyperparameter are plotted on the X-axis of the corresponding table, while the Y-axis represents the Top1 and Top5 accuracies on UCF101.**

(1) The optimal number of memory slots is 4096. Increasing it to 8192 does not enhance the performance, possibly due to the introduction of more parameters the need more training iterations or the need to adjust the scale factor $r$ in Eq. 2. (2) The best performance is achieved with 4096 coupled visual concept dictionaries. Additional dictionaries do not significantly improve performance, potentially due to the lack of adjusting the temperature coefficient or the need for longer training iterations due to the introduction of more parameters in the dictionaries. $K = 0$ means that the visual concept alignment module is not used. (3) The optimal performance is obtained when both $\tau_s$ and $\tau_t$ are set to 0.05. Smaller temperature parameters make the visual concept distributions sharper, highlighting important visual concepts for alignment while reducing the alignment of unimportant ones.

In summary, the results in Fig. 3 show that our method achieves optimal performance with 4096 memory slots, 4096 coupled visual concept dictionaries, and temperature parameters of 0.05 for both $\tau_s$ and $\tau_t$. The experiments demonstrate the effectiveness of our method under various hyperparameter settings.

## 4.4 Qualitative Analysis

In this section, we conduct a visualization analysis to examine the effectiveness of the key-value memory enhanced predictor and the visual concept alignment module. We utilize the pre-trained R3D-18 backbone as the feature extractor for this analysis. To evaluate the model’s performance in discriminating between similar action categories, we select three pairs of easily confused action categories from the UCF101 dataset: cricket bowling and cricket shooting, playing the flute and playing the violin, and pull-ups and jumping jacks. To generate visualizations, we perform a center crop on the middle segment of each video clip before feeding it into the pre-trained model.

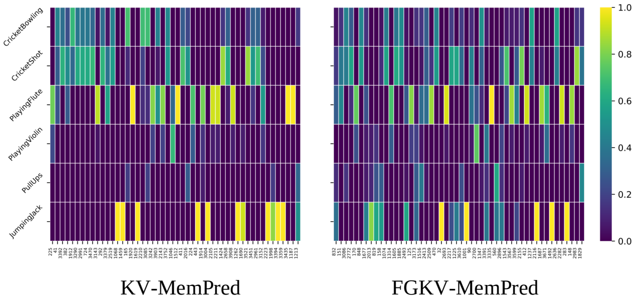


**Figure 4: Conditional probability matrices illustrating $P(\text{action}|\text{memslot})$. The Y-axis of each matrix represents the action label, while the X-axis represents the indices of the top 50 most frequently activated memory slots.**

**Analysis of the Memory Slots.** The proposed methodology incorporates a key-value memory-enhanced predictor that produces a knowledge relevance vector for each video clip. This vector contains scores representing the semantic similarity between the clip's feature and each key memory slot. We aggregate the indices of the top 20 memory slots with the highest scores for each video and associate them with the corresponding video labels. We further select the 50 most frequently occurring slots and compute the conditional probability distribution of action categories given the memory slots, denoted as $P(\text{action}|\text{memslot})$. Specifically, we aggregate action labels for each memory slot across all selected videos, allowing us to compute the conditional probability of each action given any memory slot. The resulting probabilities are depicted in Fig. 4, which reveal insightful information, including:

(1) It is observed from the distributions that easily confused categories, such as cricket bowling and cricket shooting, tend to activate the same memory slots due to the similarity in their actions and backgrounds. Furthermore, memory slots can learn shared semantic knowledge across categories, allowing the encoder to leverage closely related knowledge acquired from the entire dataset when predicting features. Notably, despite sharing many memory slots, cricket bowling, and cricket shooting still exhibit a few unique memory slots, indicating that memory slots learn category-specific semantic information to distinguish confusing categories during prediction. Similar observations can be made for other pairs of confusing categories.

(2) The comparison between KVMemPred and FGKVMemPred demonstrates the effectiveness of the visual concept alignment module. Specifically, easily confused categories like cricket bowling and cricket batting demonstrate fewer co-occurring slots and more category-specific slots. This suggests that our visual concept alignment module assists memory slots in capturing and storing more fine-grained visual concept features, thereby enhancing the ability to distinguish confusing categories. Additionally, each category activates a greater number of memory slots, indicating the proposed module enables a category to be described by more fine-grained visual concepts.

**Analysis of the Visual Concepts.** To analyze the learned dictionary, we examine the top three samples that had the highest response to the three most frequently activated visual concepts from the test set. Fig. 5 illustrates that each visual concept is associated with a specific action or scene. For example, the concept in the first row represents the action of holding a bar-shaped object with both hands, such as a barbell or oar. The concept in the second row corresponds to horse racing. The concept in the third row indicates the action of jumping. These observations confirm that each visual concept can effectively identify specific fine-grained visual concepts across diverse backgrounds and variations in human subjects, thereby validating the effectiveness of our joint dictionary learning approach.

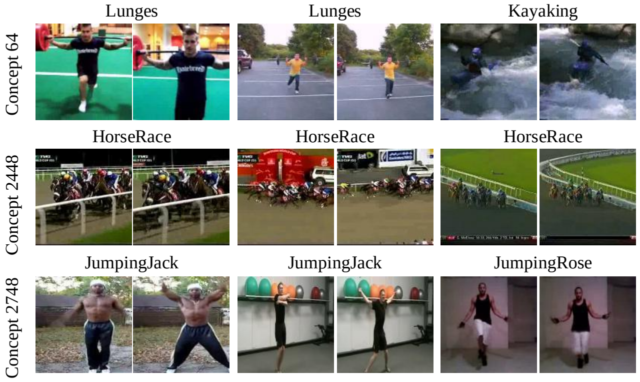


**Figure 5: Visualization of the top-3 most frequently activated visual concepts in our learned dictionaries using the UCF101 testing videos.**

## 5 CONCLUSION

In this paper, we introduce a novel self-supervised video representation learning approach that leverages the key-value memory enhanced predictor and the visual concept alignment module. By incorporating the memory enhanced predictor, our method efficiently retrieves relevant knowledge from long-term memory and effectively mitigates feature distribution gaps between encoders, thus enhancing the predictive capabilities of video representations. Additionally, the concept alignment module aligns shared visual concepts across different video views, leading to improved knowledge storage and better prediction performance. Extensive experiments demonstrate the superiority of our approach over state-of-the-art methods in action recognition and retrieval tasks on various datasets, validating the effectiveness of our method in learning generalized video representations. This research opens up exciting possibilities for exploring advanced memory-enhanced self-supervised learning techniques and innovative ways of aligning visual concepts across videos. Future work may also investigate combining different self-supervised learning paradigms to further boost video representation learning performance.

## ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (No. 62006140). Additionally, it is also supported in part by the Shandong Provincial Natural Science Foundation (No.: ZR2020QF106) and is in part based on research sponsored by Shenzhen Higher Education Institutions Stable Support Program (Key Project) under award number GXWD20220817123150002.

# REFERENCES

[1] Soheil Bahrampour, Nasser M Nasrabadi, Asok Ray, and William Kenneth Jenkins. 2015. Multimodal task-driven dictionary learning for image classification. *IEEE Transactions on Image Processing* 25, 1 (2015), 24–38.

[2] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. 2020. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems* 33 (2020), 9912–9924.

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. 2021. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE International Conference on Computer Vision*. 9650–9660.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the International Conference on Machine Learning*. PMLR, 1597–1607.

[5] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. 2020. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297* (2020).

[6] Xinlei Chen and Kaiming He. 2021. Exploring simple siamese representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 15750–15758.

[7] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[8] Ishan Dave, Rohit Gupta, Mamshad Nayeem Rizve, and Mubarak Shah. 2022. Tclr: Temporal contrastive learning for video representation. *Computer Vision and Image Understanding* 219 (2022), 103406.

[9] Shuangrui Ding, Maomao Li, Tianyu Yang, Rui Qian, Haohang Xu, Qingyi Chen, Jue Wang, and Hongkai Xiong. 2022. Motion-aware contrastive video representation learning via foreground-background merging. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 9716–9726.

[10] Shuangrui Ding, Rui Qian, and Hongkai Xiong. 2022. Dual contrastive learning for spatio-temporal representation. In *Proceedings of the 30th ACM International Conference on Multimedia*. 5649–5658.

[11] Weisheng Dong, Lei Zhang, Guangming Shi, and Xiaolin Wu. 2011. Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *IEEE Transactions on Image Processing* 20, 7 (2011), 1838–1857.

[12] Haodong Duan, Nanxuan Zhao, Kai Chen, and Dahua Lin. 2022. Transrank: Self-supervised video representation learning via ranking-based transformation recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3000–3010.

[13] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. 2019. Slowfast networks for video recognition. In *Proceedings of the IEEE International Conference on Computer Vision*. 6202–6211.

[14] Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross Girshick, and Kaiming He. 2021. A large-scale study on unsupervised spatiotemporal representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3299–3309.

[15] Cristina Garcia-Cardona and Brendt Wohlberg. 2018. Convolutional dictionary learning: a comparative review and new algorithms. *IEEE Transactions on Computational Imaging* 4, 3 (2018), 366–381.

[16] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. 2020. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems* 33 (2020), 21271–21284.

[17] Jiangfan Han, Mengya Gao, Yujie Wang, Quanquan Li, Hongsheng Li, and Xiaogang Wang. 2021. Fixing the teacher-student knowledge discrepancy in distillation. *arXiv preprint arXiv:2103.16844* (2021).

[18] Tengda Han, Weidi Xie, and Andrew Zisserman. 2019. Video representation learning by dense predictive coding. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 0–0.

[19] Tengda Han, Weidi Xie, and Andrew Zisserman. 2020. Memory-augmented dense predictive coding for video representation learning. In *Proceedings of the European Conference on Computer Vision*.

[20] Tengda Han, Weidi Xie, and Andrew Zisserman. 2020. Self-supervised co-training for video representation learning. *Advances in Neural Information Processing Systems* 33 (2020), 5679–5690.

[21] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. 2017. Learning spatio-temporal features with 3d residual networks for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision) workshops*. 3154–3160.

[22] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. 2018. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6546–6555.

[23] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 9729–9738.

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

[25] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[26] De-An Huang and Yu-Chiang Frank Wang. 2013. Coupled dictionary and feature space learning with applications to cross-domain image synthesis and recognition. In *Proceedings of the IEEE International Conference on Computer Vision*. 2496–2503.

[27] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*. pmlr, 448–456.

[28] Simon Jenni and Hailin Jin. 2021. Time-equivariant contrastive video representation learning. In *Proceedings of the IEEE International Conference on Computer Vision*. 9970–9980.

[29] Simon Jenni, Givi Meishvili, and Paolo Favaro. 2020. Video representation learning by recognizing temporal transformations. In *Proceedings of the European Conference on Computer Vision*.

[30] Yeong Jun Koh and Chang-Su Kim. 2017. Primary object segmentation in videos based on region augmentation and reduction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3442–3450.

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

[32] Dahun Kim, Donghyeon Cho, and In So Kweon. 2019. Self-supervised video representation learning with space-time cubic puzzles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 8545–8552.

[33] Minsu Kim, Joanna Hong, Se Jin Park, and Yong Man Ro. 2021. Cromm-vsr: Cross-modal memory augmented visual speech recognition. *IEEE Transactions on Multimedia* 24 (2021), 4342–4355.

[34] Minsu Kim, Joanna Hong, Se Jin Park, and Yong Man Ro. 2021. Multi-modality associative bridging through memory: Speech sound recollected from face video. In *Proceedings of the IEEE International Conference on Computer Vision*. 296–306.

[35] Haofei Kuang, Yi Zhu, Zhi Zhang, Xinyu Li, Joseph Tighe, Sören Schwertfeger, Cyrill Stachniss, and Mu Li. 2021. Video contrastive learning with global context. In *Proceedings of the IEEE International Conference on Computer Vision*. 3195–3204.

[36] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. 2011. HMDB: a large video database for human motion recognition. In *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, 2556–2563.

[37] Yuanze Lin, Xun Guo, and Yan Lu. 2021. Self-supervised video representation learning with meta-contrastive network. In *Proceedings of the IEEE International Conference on Computer Vision*. 8239–8249.

[38] Xiankai Lu, Wenguan Wang, Martin Danelljan, Tianfei Zhou, Jianbing Shen, and Luc Van Gool. 2020. Video object segmentation with episodic graph memory networks. In *Proceedings of the European Conference on Computer Vision*.

[39] Dezhao Luo, Chang Liu, Yu Zhou, Dongbao Yang, Can Ma, Qixiang Ye, and Weiping Wang. 2020. Video cloze procedure for self-supervised spatio-temporal learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 11701–11708.

[40] Zelun Luo, Boya Peng, De-An Huang, Alexandre Alahi, and Li Fei-Fei. 2017. Unsupervised learning of long-term motion dynamics for videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2203–2212.

[41] Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. *arXiv preprint arXiv:1606.03126* (2016).

[42] Liqiang Nie, Leigang Qu, Dai Meng, Min Zhang, Qi Tian, and Alberto Del Bimbo. 2022. Search-oriented micro-video captioning. In *Proceedings of the 30th ACM International Conference on Multimedia*. 3234–3243.

[43] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[44] Tian Pan, Yibing Song, Tianyu Yang, Wenhao Jiang, and Wei Liu. 2021. Videomoco: Contrastive video representation learning with temporally adversarial examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 11205–11214.

[45] Hyunjong Park, Jongyoun Noh, and Bumsub Ham. 2020. Learning memory-guided normality for anomaly detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 14372–14381.

[46] Jungin Park, Jiyoung Lee, Ig-Jae Kim, and Kwanghoon Sohn. 2022. Probabilistic representations for video contrastive learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 14711–14721.

[47] Trung X Pham, Axi Niu, Zhang Kang, Sultan Rizky Madjid, Ji Woo Hong, Daehyeok Kim, Joshua Tian Jin Tee, and Chang D Yoo. 2022. Self-Supervised Visual Representation Learning via Residual Momentum. *arXiv preprint arXiv:2211.09861* (2022).

[48] Rui Qian, Shuangrui Ding, Xian Liu, and Dahua Lin. 2022. Static and dynamic concepts for self-supervised video representation learning. In *Proceedings of the European Conference on Computer Vision*.

[49] Rui Qian, Yuxi Li, Huabin Liu, John See, Shuangrui Ding, Xian Liu, Dian Li, and Weiyao Lin. 2021. Enhancing self-supervised video representation learning via multi-level feature optimization. In *Proceedings of the IEEE International*

Conference on Computer Vision. 7990–8001.

[50] Rui Qian, Weiyao Lin, John See, and Dian Li. 2022. Controllable Augmentations for Video Representation Learning. *arXiv preprint arXiv:2203.16632* (2022).

[51] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. 2021. Spatiotemporal contrastive video representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6964–6974.

[52] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2014. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550* (2014).

[53] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402* (2012).

[54] Hillel Sreter and Raja Giryes. 2018. Learned convolutional sparse coding. In *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2191–2195.

[55] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. *Advances in Neural Information Processing Systems* 28 (2015).

[56] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1–9.

[57] Bahareh Tolooshams, Sourav Dey, and Demba Ba. 2020. Deep residual autoencoders for expectation maximization-inspired dictionary learning. *IEEE Transactions on Neural Networks and Learning Systems* 32, 6 (2020), 2415–2429.

[58] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. 2022. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in neural information processing systems* 35 (2022), 10078–10093.

[59] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. 2018. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6450–6459.

[60] Jiangliu Wang, Jianbo Jiao, and Yun-Hui Liu. 2020. Self-supervised video representation learning by pace prediction. In *Proceedings of the European Conference on Computer Vision*.

[61] Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *arXiv preprint arXiv:1410.3916* (2014).

[62] Jianlong Wu, Wei Sun, Tian Gan, Ning Ding, Feijun Jiang, Jialie Shen, and Liqiang Nie. 2023. Neighbor-Guided Consistent and Contrastive Learning for Semi-Supervised Action Recognition. *IEEE Transactions on Image Processing* (2023).

[63] Huaxin Xiao, Bingyi Kang, Yu Liu, Maojun Zhang, and Jiashi Feng. 2019. Online meta adaptation for fast video object segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, 5 (2019), 1205–1217.

[64] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. 2019. Self-supervised spatiotemporal learning via video clip order prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 10334–10343.

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

[66] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. 2010. Image super-resolution via sparse representation. *IEEE Transactions on Image Processing* 19, 11 (2010), 2861–2873.

[67] Yang You, Igor Gitman, and Boris Ginsburg. 2017. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888* (2017).

[68] Yujia Zhang, Lai-Man Po, Xuyuan Xu, Mengyang Liu, Yexin Wang, Weifeng Ou, Yuzhi Zhao, and Wing-Yin Yu. 2022. Contrastive spatio-temporal pretext learning for self-supervised video representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 3380–3389.

[69] Yuxin Zhang, Jindong Wang, Yiqiang Chen, Han Yu, and Tao Qin. 2022. Adaptive memory networks with self-supervised learning for unsupervised anomaly detection. *IEEE Transactions on Knowledge and Data Engineering* (2022).

[70] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. 2017. Temporal action detection with structured segment networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 2914–2923.

[71] Hongyi Zheng, Hongwei Yong, and Lei Zhang. 2021. Deep convolutional dictionary learning for image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 630–641.

# A ILLUSTRATION OF THE FEATURE DISTRIBUTION GAP

Similar to BYOL [16] and DINO [3], which view self-supervised learning as a form of knowledge distillation [17, 24, 47], our approach aims to minimize the discrepancy between student and teacher feature spaces. However, instead of mimicking intermediate layer features as in traditional distillation methods [52], we introduce key-value memory networks to enhance the encoder's prediction capability.

To validate the effectiveness of our fine-grained key-value memory enhanced predictor (FGKVMemPred) in narrowing the distribution gap, we compute the negative cosine similarity between features from the online encoder ($y_{\text{src}}$) and the target encoder ($y_{\text{tgt}}$) and visualize the similarity distributions in Fig. 6. The results demonstrate that FGKVMemPred significantly reduces the feature distribution gap compared to the baseline model using the MLP predictor. This reduction in the distribution gap indicates the improved alignment of features between the online and target encoders, further validating the effectiveness of our proposed method.

# B CONNECTION AND DIFFERENCE BETWEEN MEMORIES AND DICTIONARIES

Memory slots and dictionaries in our framework share certain similarities while exhibiting distinct characteristics:

**Similarities**

- Both operate in a QKV transformer-style manner, employing representations from different network levels as *queries*.
- They function as lifelong memories, accumulating knowledge throughout the training process.

**Differences**

- **Functionality**: **Memory slots** serve as a comprehensive knowledge repository, enabling the model to access relevant knowledge from all slots for prediction. This facilitates leveraging a wide range of stored information from the entire dataset. **Dictionaries**, on the other hand, are designed to align fine-grained semantic information, enhancing the richness of knowledge stored in memory slots, particularly in the context of videos. Their role is critical in aligning visual concepts across diverse views and temporal segments.
- **Structure**: **Memory slots** are an integral part of the key-value memory network and represent stored knowledge or high-level concepts derived from the entire video dataset. They capture essential information learned from various video clips, facilitating knowledge transfer across different instances. **Dictionaries** consist of collections of visual concepts associated with each encoder. Their main purpose is to support the alignment of fine-grained semantic information within the same video, capturing nuanced relationships and video-specific visual patterns.
- **Supervision**: **Memory slots** are supervised by the prediction loss, utilizing input features from the online encoder and the prediction targets from the target encoder. This supervision enables the model to predict features from temporally different video clips, incorporating both the input context features and the retrieved stored knowledge from the entire dataset. **Dictionaries** follow a coupled dictionary learning [26] design. Each dictionary functions as an autoencoder-decoder [54], utilizing input features and reconstruction targets from the same encoder. Through the reconstruction of input features using coding vectors, dictionaries facilitate the alignment of fine-grained visual concepts across different views within the same video. The joint optimization of

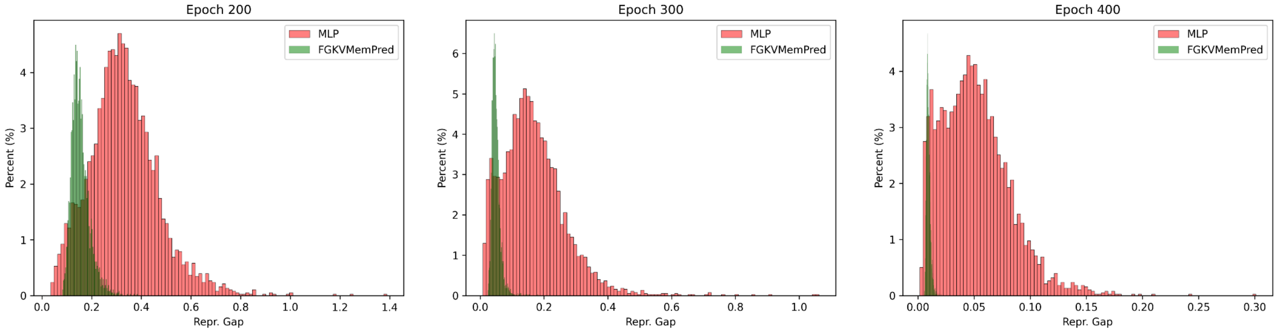


**Figure 6: Visualization of Feature Distribution Gaps.**

the reconstruction loss and alignment loss further enhances the model's ability to capture and align relevant visual information.

The contributions of the memories and dictionaries are presented in Tab. 4 and Fig. 4 of our paper, which also illustrate their distinct functionalities.

## C ABLATION STUDY OF $\alpha$ AND $\beta$ IN EQ. 9

We conducted an ablation study to examine the impact of different $\alpha$ and $\beta$ values on the model's performance. The summarized results are presented in the following tables. In these experiments, the R3D-18 backbone is employed, and the training and evaluation settings correspond to those outlined in Table 4.

**Table 5: Effect of Different $\alpha$ Values with $\beta = 0.01$**

| $\alpha$ | **0** | **0.002** | **0.01** | **0.05** |
|---|---|---|---|---|
| UCF101 | 67.29 | 67.33 | 68.89 | 68.94 |
| HMDB51 | 36.05 | 34.67 | 36.45 | 36.45 |

**Table 6: Effect of Different $\beta$ Values with $\alpha = 0.01$**

| $\beta$ | **0** | **0.002** | **0.01** | **0.05** |
|---|---|---|---|---|
| UCF101 | 68.91 | 69.81 | 68.89 | 68.36 |
| HMDB51 | 36.91 | 37.96 | 36.45 | 36.97 |

Setting $\alpha$ to 0 results in reduced performance, highlighting the significance of the $\mathcal{L}_{\text{recon}}$ loss. Slightly increasing $\alpha$ improves performance, indicating its positive impact. Setting $\beta$ to 0 does not lead to a significant performance drop, indicating that the reconstruction supervision can enhance the online encoder's performance [58]. Additionally, increasing $\beta$ results in a notable performance increase, underscoring the importance of $\mathcal{L}_{\text{align}}$.

These findings underscore the crucial roles played by both losses. Properly balancing $\alpha$ and $\beta$ enhances the model's performance. For the UCF101 and HMDB51 datasets, optimal values are $\alpha = 0.01$ and $\beta = 0.002$. These ablation experiments will be incorporated into the paper.