Supplementary Materials for HiCo

Zhiwu Qing[1] Shiwei Zhang[2*] Ziyuan Huang[3] Yi Xu[4] Xiang Wang[1]
Mingqian Tang[2] Changxin Gao[1*] Rong Jin[2] Nong Sang[1]
[1]Key Laboratory of Image Processing and Intelligent Control
School of Artificial Intelligence and Automation, Huazhong University of Science and Technology
[2]Alibaba Group [3]ARC, National University of Singapore [4]Dalian Unversity of Technology
{qzw, wxiang, cgao, nsang}@hust.edu.cn
{zhangjin.zsw, mingqian.tmq, jinrong.jr}@alibaba-inc.com
ziyuan.huang@u.nus.edu yxu@dlut.edu.cn

Overview

In this supplementary material, we first provide detailed theoretical proof for our proposed Gradual Sampling in Section 1. Then the implementation details for pre-training, action recognition, video retrieval, and temporal action localization are introduced in Section 2. Finally, we also show more experimental results and visualizations in Section 3.

1. Theoretical Analysis of the Gradual Sampling Strategy

In this section, we provide a theoretical understanding of the proposed Gradual Sampling (GS) strategy from the view of generalization analysis, which is commonly used in the literature of learning theory [13]. For the sake of simplicity of analysis, we abstract the key points from the GS strategy and make the strategy more math-friendly. As mentioned in the main content of this paper, we divide the training data into two groups, one with small variance (denoted by $\widehat{\mathcal{D}}_s$) and another one with large variance (denoted by $\widehat{\mathcal{D}}_l$). At the beginning of training, the sampled clips are considered as examples with small variance since its sampling window size is small according to the definition of $\hat{\Delta}_{\text{max}}(\alpha)$. This is reasonable because when the window size is small, the sampled clips are usually similar. While during the later training epochs, the sampled clips could be examples either with large or with small variance since the sampling window size is large and thus it could sample very different clips. From the viewpoint of optimization, we characterize the difficulty of examples by their variance in gradients. For instance, given two types of examples that are sampled from two different distributions $\mathcal{D}_s$ and $\mathcal{D}_l$, it is easy to learn a prediction function from $\mathcal{D}_s$ than from $\mathcal{D}_l$, if

$$\mathrm{E}_{\xi\sim\mathcal{D}_s}\left[\|\nabla\ell(w;\xi)-\nabla\mathcal{F}_s(w)\|^2\right]\leq\mathrm{E}_{\zeta\sim\mathcal{D}_l}\left[\|\nabla\ell(w;\zeta)-\nabla\mathcal{F}_l(w)\|^2\right], \tag{A1}$$

where ξ and ζ are the data examples, w is the model parameter, ℓ the loss function and

$$\nabla\mathcal{F}_s(w)=\mathrm{E}_{\xi\sim\mathcal{D}_s}\left[\nabla\ell(w;\xi)\right],\quad \nabla\mathcal{F}_l(w)=\mathrm{E}_{\zeta\sim\mathcal{D}_l}\left[\nabla\ell(w;\zeta)\right]. \tag{A2}$$

In the remaining of this section, we first give the preliminary, then we present the main result in a theorem. All the proofs are included at the end of this section.

1.1. Preliminary

To make it easy for our analysis, we formulate the target task as a optimization problem as follows, where the target distribution is a mixture of distributions $\mathcal{D}_s$ and $\mathcal{D}_l$, with a mixture probability $p\in[0,1]$. Formally, the optimization is defined as

$$\min_{w\in\mathbb{R}^d}\mathcal{L}(w):=(1-p)\mathcal{F}_s(w)+p\mathcal{F}_l(w), \tag{A3}$$

where w is the model parameter to be learned, and the loss functions for simple examples and difficult examples are respectively given by

$$\mathcal{F}_s(w) = \mathrm{E}_{\xi\sim\mathcal{D}_s}\left[\ell(w;\xi)\right], \quad \mathcal{F}_l(w) = \mathrm{E}_{\zeta\sim\mathcal{D}_l}\left[\ell(w;\zeta)\right]. \tag{A4}$$

Here the data examples ξ and ζ are the data examples that follow distributions $\mathcal{D}_s$ and $\mathcal{D}_l$ respectively, ℓ is a general loss function that can be a single loss or a combined loss of several loss functions. The problem (A3) is known as risk minimization (RM). Since the distributions $\mathcal{D}_s$ and $\mathcal{D}_l$ are usually unknown, it is difficult to obtain the loss function $\mathcal{L}(w)$ explicitly. In stead of RM, one can consider its empirical version, which is known an empirical risk minimization (ERM):

$$\min_{w\in\mathbb{R}^d} \widehat{\mathcal{L}}(w) := (1-p)\widehat{\mathcal{F}}_s(w) + p\widehat{\mathcal{F}}_l(w), \tag{A5}$$

where

$$\widehat{\mathcal{F}}_s(w) = \frac{1}{n}\sum_{\xi_i\in\widehat{\mathcal{D}}_s} \ell(w;\xi_i), \quad \widehat{\mathcal{F}}_l(w) = \frac{1}{m}\sum_{\zeta_j\in\widehat{\mathcal{D}}_l} \ell(w;\zeta_j). \tag{A6}$$

The set of training data $\widehat{\mathcal{D}}_s := \{\xi_i, i = 1,\dots,n\}$ is sampled from the distribution $\mathcal{D}_s$, and the set of training data $\widehat{\mathcal{D}}_l := \{\zeta_j, i = 1,\dots,m\}$ is sampled from the distribution $\mathcal{D}_l$. To solve the ERM (A5), one of simple yet efficient methods is SGD, whose key updating step is given by

$$w_{t+1} = w_t - \eta\nabla_w g(w_t;\xi_{i_t},\zeta_{i_t}), \ \ t = 0,1,\dots, \tag{A7}$$

where $\eta > 0$ is the learning rate, and $\nabla_w g(w;\xi,\zeta)$ the stochastic gradient of $\mathcal{L}(w)$ such that $\mathrm{E}_{\xi,\zeta}[\nabla g(w;\xi,\zeta)] = \nabla\mathcal{L}(w)$. For simplicity, we use $g(w) := g(w;\xi,\zeta)$ in the following analysis. When the variable to be taken a gradient is obvious, we use $\nabla g(w)$ instead of $\nabla_w g(w)$. Similarly, when the randomness is obvious, we use $\mathrm{E}[\cdot]$ instead of $\mathrm{E}_\xi[\cdot]$, $\mathrm{E}_\zeta[\cdot]$ or $\mathrm{E}_{\xi,\zeta}[\cdot]$. In this analysis, we are interested in the excess risk bound (ERB), which is a standard measurement of evaluating the solution $\widehat{w}$ obtained by an algorithm:

$$\mathcal{L}(\widehat{w}) - \mathcal{L}(w_*), \tag{A8}$$

where $w_* \in \arg\min_{w\in\mathbb{R}^d}\mathcal{L}(w)$ is the optimal solution of problem (A3).

For the convenience of analysis, we make the following widely used assumptions for the loss function.

Assumption 1 (Polyak-Łojasiewicz condition [10])**.** *There exists a constant $\mu > 0$ such that*

$$2\mu(\mathcal{L}(w) - \mathcal{L}(w_*)) \le \|\nabla\mathcal{L}(w)\|^2, \quad \forall w\in\mathbb{R}^d,$$

where $w_* \in \arg\min_{w\in\mathbb{R}^d}\mathcal{L}(w)$ is a optimal solution.

Assumption 2 (Smoothness [9])**.** *$\mathcal{L}(w)$ is smooth with an L-Lipchitz continuous gradient, i.e., it is differentiable and there exists a constant $L > 0$ such that*

$$\|\nabla\mathcal{L}(w) - \nabla\mathcal{L}(w')\| \le L\|w - w'\|, \forall w, w'\in\mathbb{R}^d.$$

Assumption 2 says the objective function $\mathcal{L}(w)$ is smooth with module parameter $L > 0$. This assumption has an equivalent expression [9]: $\mathcal{L}(w) - \mathcal{L}(w') \le \langle\mathcal{L}(w'), w - w'\rangle + \frac{L}{2}\|w - w'\|^2, \quad \forall w, w'\in\mathbb{R}^d$.

We assume that the difference between $\mathcal{F}_s$ and $\mathcal{F}_l$ is captured by $\hat{\Delta}$ from the following assumption.

Assumption 3. *There exists $\hat{\Delta} \ge 0$ such that*

$$\max_{w\in\mathbb{R}^d} \|\nabla\mathcal{F}_s(w) - \nabla\mathcal{F}_l(w)\| \le \hat{\Delta}.$$

Following the above definition of difficult examples, we assume the following variance structure for stochastic gradients for distribution $\mathcal{D}_s$ and $\mathcal{D}_l$.

Assumption 4 (Bounded variance [4]). *The stochastic gradient of $\mathcal{F}_s(w)$ is unbiased and variance bounded. That is, $\mathrm{E}_{\xi\sim\mathcal{D}_s}[\nabla\ell(w;\xi)] = \nabla\mathcal{F}_s(w)$ and there exists a constant $\sigma^2 > 0$, such that*

$$\mathrm{E}_{\xi\sim\mathcal{D}_s}\left[\|\nabla\ell(w;\xi) - \nabla\mathcal{F}_s(w)\|^2\right] \leq \sigma^2.$$

Assumption 5 (Weak Growth Condition [1, 2]). *The stochastic gradient of $\mathcal{L}(w)$ is unbiased and variance bounded. That is, $\mathrm{E}_{\xi\sim\mathcal{D}_s}\mathrm{E}_{\zeta\sim\mathcal{D}_l}[\nabla g(w)] = \nabla\mathcal{L}(w)$ and there exists a constant $\sigma^2 > 0$, such that*

$$\mathrm{E}_{\xi\sim\mathcal{D}_l}\mathrm{E}_{\zeta\sim\mathcal{D}_l}\left[\|\nabla g(w) - \nabla\mathcal{L}(w)\|^2\right] \leq \frac{h}{2}\|\nabla\mathcal{L}(w)\|^2 + \sigma^2,$$

where $h \gg 1$ is a large constant.

Assumptions 4 and 5 imply that $\mathrm{E}_{\zeta\sim\mathcal{D}_l}[\nabla\ell(w;\zeta)] = \nabla\mathcal{F}_l(w)$ and

$$\mathrm{E}_{\xi\sim\mathcal{D}_l}\left[\|\nabla\ell(w;\zeta) - \nabla\mathcal{F}_l(w)\|^2\right] \leq D + \sigma^2, \tag{A9}$$

where $D := \frac{h}{2p^2}\|\nabla\mathcal{L}(w)\|^2 + \frac{(1-p)^2}{p^2}\left(\sigma^2 - E_{\xi\sim\mathcal{D}_s}\left[\|\nabla\ell(w;\xi) - \nabla\mathcal{F}_s(w)\|^2\right]\right) > 0$. Please note that h is a very large constant, thus we can consider that the variance for difficult examples is much larger than the variance for simple examples.

As indicated by the variance structures, stochastic gradients from $\mathcal{D}_l$ exhibit significantly larger variance than those from $\mathcal{D}_s$, particularly at the beginning of the optimization. Hence, it may not be a good idea to run the standard SGD to optimize $\mathcal{L}(w)$. Instead, we could divide the training process into two phases. In the first phase, we will optimize $\mathcal{L}(w)$ using the SGD using the easy examples sampled from distribution $\mathcal{D}_s$. In this way, we could avoid the potentially variance arising from $\mathcal{D}_l$, of course, at the price of bias. In the second phase, when we already received a good solution, we will run the standard SGD to optimize $\mathcal{L}(w)$. Since the solution received from phase I already has excess risk, we will not suffer from the large variance arising from distribution $\mathcal{D}_l$.

1.2. Theoretical Analysis

Before the mathematically analysis, we give the following formal version of Theorem 1, showing that the proposed GS strategy has better generalization than the random sampling (RS) strategy under some mild assumptions.

Theorem 1 (Formal Version of Theorem 1). *Under Assumptions 1, 2, 3, 4, 5, we have the following two ERB for RS and GS, respectively.*
(1) for the output of RS $\widehat{w}_{rs}$, by setting the learning rate $\eta \leq 1/[L(1+hp)]$, the we have

$$\mathrm{E}\left[\mathcal{L}(\widehat{w}_{rs}) - \mathcal{L}(w_*)\right] \leq \exp(-\eta\mu(n+m))(\mathcal{L}(w_0) - \mathcal{L}(w_*)) + \frac{\eta L\sigma^2}{2\mu} \leq O\left(\mathcal{L}(w_0) - \mathcal{L}(w_*)\right).$$

(2) for the output of GS $\widehat{w}_{gs}$, by setting the learning rates $\eta_1 = 1/L$ in the first phase and $\eta \leq 1/[L(1+hp)]$ in the second phase, the we have

$$\mathrm{E}\left[\mathcal{L}(\widehat{w}_{gs}) - \mathcal{L}(w_*)\right] \leq O\left(\frac{\sigma^2 L\log(n)}{\mu^2 n} + \frac{p^2\hat{\Delta}^2}{\mu}\right).$$

1.2.1 Proof of Theorem 1 (1)

As the first step, we analyze the RS for optimizing $\mathcal{L}(w)$, where uses both the examples from $\mathcal{D}_s$ and $\mathcal{D}_l$.

Proof. For the sake of simplicity, let denote by $\nabla g(w)$ the stochastic gradient of $\mathcal{L}(w)$ such that $\mathrm{E}[\nabla g(w)] = \nabla\mathcal{L}(w)$. Then the update of SGD for $w_{t+1} = w_t - \eta\nabla g(w_t)$ for $t = 0, 1, 2, \ldots$. By the smoothness of function $\mathcal{L}$ from Assumption 2, we

have

$$\begin{aligned}
&\mathrm{E}[\mathcal{L}(w_{t+1}) - \mathcal{L}(w_t)] \\
\leq &\mathrm{E}[\langle w_{t+1} - w_t, \nabla\mathcal{L}(w_t)\rangle] + \frac{L}{2}\mathrm{E}[\|w_{t+1} - w_t\|^2] \\
= &-\eta\mathrm{E}[\langle \nabla g(w_t), \nabla\mathcal{L}(w_t)\rangle] + \frac{\eta^2 L}{2}\mathrm{E}[\|\nabla g(w_t)\|^2] \\
= &-\eta\left(1 - \frac{\eta L}{2}\right)\|\nabla\mathcal{L}(w_t)\|^2 + \frac{\eta^2 L}{2}\mathrm{E}[\|\nabla g(w_t) - \nabla\mathcal{L}(w_t)\|^2] \\
\leq &-\eta\left(1 - \frac{\eta L}{2}\right)\|\nabla\mathcal{L}(w_t)\|^2 + \frac{\eta^2 L}{2}\mathrm{E}\left[\frac{h}{2}\|\nabla\mathcal{L}(w_t)\|^2 + \sigma^2\right],
\end{aligned} \tag{A10}$$

where the last inequality uses Assumptions 4 and 5. Due to Assumption 1,

$$\mathrm{E}[\mathcal{L}(w_{t+1}) - \mathcal{L}(w_t)] \leq -\eta\left(1 - \frac{\eta L(1+h)}{2}\right)\|\nabla\mathcal{L}(w_t)\|^2 + \frac{\eta^2 L\sigma^2}{2}. \tag{A11}$$

By selecting $\eta \leq \eta_* := 1/[L(1+h)]$, we have

$$\mathrm{E}\left[\mathcal{L}(w_{t+1}) - \mathcal{L}(w_*)\right] \leq (1 - \eta\mu)\,\mathrm{E}\left[\mathcal{L}(w_t) - \mathcal{L}(w_*)\right] + \frac{\eta^2 L\sigma^2}{2}. \tag{A12}$$

and therefore

$$\mathrm{E}\left[\mathcal{L}(w_{n+m+1}) - \mathcal{L}(w_*)\right] \leq \exp(-\eta\mu(n+m))(\mathcal{L}(w_0) - \mathcal{L}(w_*)) + \frac{\eta L\sigma^2}{2\mu}. \tag{A13}$$

□

Remark Since h is large enough, implying that $\eta_* := 1/[L(1+hp)]$ is small enough, such that

$$\exp(-\eta_*\mu(n+m)) \geq \frac{L\sigma^2}{2\mu^2(n+m)(\mathcal{L}(w_0) - \mathcal{L}(w_*))},$$

we have

$$\mathrm{E}\left[\mathcal{L}(w_{n+m+1}) - \mathcal{L}(w_*)\right] \leq \exp(-\eta_*\mu(n+m))(\mathcal{L}(w_0) - \mathcal{L}(w_*)) + \frac{\eta_* L\sigma^2}{2\mu}.$$

Consider the special case when $n + m = (hp+1)\kappa$ where $\kappa := L/\mu$ (now $\eta_* = \frac{1}{(n+m)\mu}$), and

$$e^{-1} \geq \frac{\sigma^2}{2\mu h\left(\mathcal{L}(w_0) - \mathcal{L}(w_*)\right)}$$

we have

$$\mathrm{E}\left[\mathcal{L}(w_{n+m+1}) - \mathcal{L}(w_*)\right] \leq \frac{\mathcal{L}(w_0) - \mathcal{L}(w_*)}{e} + \frac{\sigma^2}{2\mu(h+1)}.$$

We can see that, due to the large variance arising from $\mathcal{D}_l$, we did not receive a significant reduction in the objective even after $n + m$ iterations when applying RS strategy.

1.2.2 Proof of Theorem 1 (2)

Proof. In the first phase, we run the optimization using the examples sampled from the distribution $\mathcal{D}_s$. For the sake of simplicity, let the training examples $\xi_{i_t}, i_t = 1, \ldots, n'$ are sampled from distribution $\mathcal{D}_s$. Then the update of SGD for

$w_{t+1} = w_t - \eta_1 \nabla \ell(w_t; \xi_{i_t})$ for $i_t = 0, 1, 2, \ldots$. By the smoothness of function $\mathcal{L}$ from Assumption 2, we have

$$\begin{aligned}
&\mathrm{E}[\mathcal{L}(w_{t+1}) - \mathcal{L}(w_t)] \\
\leq &\mathrm{E}[\langle w_{t+1} - w_t, \nabla \mathcal{L}(w_t)\rangle] + \frac{L}{2}\mathrm{E}[\|w_{t+1} - w_t\|^2] \\
= &-\eta_1 \mathrm{E}[\langle \nabla \ell(w_t; \xi_{i_t}), \nabla \mathcal{L}(w_t)\rangle] + \frac{\eta_1^2 L}{2}\mathrm{E}[\|\nabla \ell(w_t; \xi_{i_t})\|^2] \\
= &\frac{\eta_1}{2}\|\nabla \mathcal{F}_s(w_t) - \nabla \mathcal{L}(w_t)\|^2 - \frac{\eta_1}{2}\|\nabla \mathcal{L}(w_t)\|^2 - \frac{\eta_1(1-\eta_1 L)}{2}\mathrm{E}[\|\nabla \mathcal{F}_s(w_t)\|^2] \\
&+ \frac{\eta_1^2 L}{2}\mathrm{E}[\|\nabla \ell(w_t; \xi_{i_t}) - \nabla \mathcal{F}_s(w_t)\|^2]. \qquad \text{(A14)}
\end{aligned}$$

where the last inequality uses $\mathrm{E}[\nabla g(w_t; \xi_t)] = \nabla g(w_t)$. Due to Assumptions 3, 4, problem definition A3 and $\eta_1 \leq 1/L$, we have

$$\mathrm{E}[\mathcal{L}(w_{t+1}) - \mathcal{L}(w_t)] \leq \frac{\eta_1 p^2 \hat{\Delta}^2}{2} + \frac{\eta_1^2 \sigma^2 L}{2} - \frac{\eta_1}{2}\|\nabla \mathcal{L}(w_t)\|^2 \qquad \text{(A15)}$$

Since $\mathcal{L}(\cdot)$ is a μ-PL function under Assumption 1, we have

$$\mathrm{E}[\mathcal{L}(w_{t+1}) - \mathcal{L}(w_t)] \leq -\eta_1 \mu \mathrm{E}[(\mathcal{L}(w_t) - \mathcal{L}(w_*))] + \frac{\eta_1 p^2 \hat{\Delta}^2}{2} + \frac{\eta_1^2 \sigma^2 L}{2} \qquad \text{(A16)}$$

and thus

$$\mathrm{E}[\mathcal{L}(w_{n'+1}) - \mathcal{L}(w_*)] \leq \exp\left(-\eta_1 \mu n'\right)\left(\mathcal{L}(w_0) - \mathcal{L}(w_*)\right) + \frac{p^2 \hat{\Delta}^2}{2\mu} + \frac{\eta_1 \sigma^2 L}{2\mu}. \qquad \text{(A17)}$$

In the second phase, we analyze the standard SGD for optimizing $\mathcal{L}(w)$ by using the solution of the first phase as the initial solution of SGD. The proof is similar to proof of Theorem 1 (2). By using the result of (A17), we have

$$\mathrm{E}\left[\mathcal{L}(w_{n+m+1}) - \mathcal{L}(w_*)\right] \leq \exp(-\eta\mu n'')\left(\exp\left(-\eta_1 \mu n'\right)\left(\mathcal{L}(w_0) - \mathcal{L}(w_*)\right) + \frac{p^2 \hat{\Delta}^2}{2\mu} + \frac{\eta_1 \sigma^2 L}{2\mu}\right) + \frac{\eta L \sigma^2}{2\mu}.$$

When $\hat{\Delta} = 0$ (i.e. the gradients for simple examples and for difficult examples are same), since $\eta = O(1/h)$ is very small, then by letting $\eta_1 = \frac{1}{\mu n'} \log\left(\frac{2\mu^2 n'(\mathcal{L}(w_0) - \mathcal{L}(w_*))}{\sigma^2 L}\right) \leq 1/L$ with $n' = n$, we have

$$\mathrm{E}\left[\mathcal{L}(w_{n+m+1}) - \mathcal{L}(w_*)\right] \leq O\left(\frac{\sigma^2 L \log(n)}{\mu^2 n}\right).$$

When $\hat{\Delta} \neq 0$ (i.e. the gradients for simple examples and for difficult examples are not same), since $\eta = O(1/h)$ is very small, then by letting $\eta_1 = \min\left(1/L, p^2\hat{\Delta}^2/(2\sigma^2 L)\right)$ and $n' \geq \frac{1}{\eta_1 \mu} \log\left(\frac{4\mu(\mathcal{L}(w_0) - \mathcal{L}(w_*))}{p^2 \hat{\Delta}^2}\right)$, we have

$$\mathrm{E}\left[\mathcal{L}(w_{n+m+1}) - \mathcal{L}(w_*)\right] \leq O\left(\frac{p^2 \hat{\Delta}^2}{\mu}\right).$$

□

2. Implementation Details

2.1. Pre-training

We pre-train all models based on the SimCLR [3] framework and set $\tau = 0.1$. Three different architectures (S3D-G, R(2+1)D-10, R3D-18) are employed as the encoder f. The visual projection head g and topical projection head h each have two hidden layers with 128 output dimensions. In addition, the topical predictor ϕ also contains two hidden layers, while

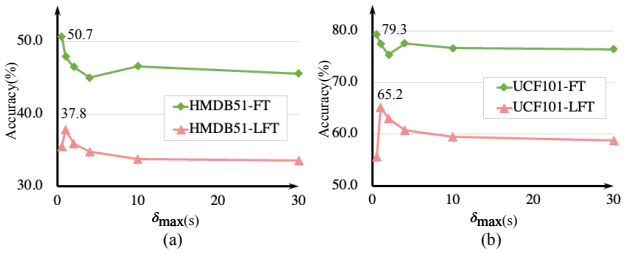

Figure A1. Different δ_{max}, *i.e.*, the maximum distance between two sampled clips for visual consistency learning. The backbone is S3D-G.

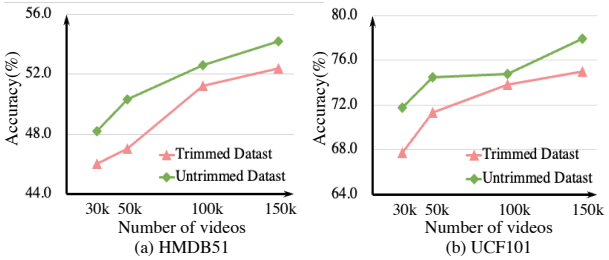

Figure A2. Linear fine-tuning performance comparisons with different numbers of videos for pre-training based on HiCo. S3D-G is employed as the backbone here.

the output dimension is 1 for topical consistency prediction. The hyperparameter of Focal Loss in $\mathcal{L}_{\text{TP}}$, *i.e.*, γ, is set to 0.5. We adopt standard augmentations used in contrastive approaches for data transformations, *e.g.*, random cropping, color distortion, and horizontal flipping. We use LARS [15] optimizer and set the base learning rate to 0.3. The training learning rate satisfies: $\texttt{LearningRate} = 0.3 \times \texttt{BatchSize}/256$. In pre-training, the learning rate is first linearly increased to $\texttt{LearningRate}$ and then decayed with the cosine schedule without restarts. The batch size is respectively set to 1024, 512, and 1024 for S3D-G, R(2+1)D-10, and R3D-18 networks. For saving the computational costs, each input clip contains 16 frames for S3D-G and R(2+1)D-10, and 8 frames for R3D-18. The spatial resolution is 112×112. For ablation experiments, we only train 120 and 50 epochs on HACS and UK400 datasets, respectively. Our final reported performances are pre-trained for 600 and 500 epochs on these two datasets, respectively.

2.2. Action Recognition

We fine-tune the models pre-trained by HiCo on UCF101 and HMDB51. If not specified, the input size of the video clips is set to 16×112×112, which is consistent with the pre-training stage. For optimizer, we utilize Adam [6] with batch sizes 1024, 256, 128 for S3D-G [14], R(2+1)D-10 [11], and R3D18 [5], respectively. The learning rate for these three backbones is set to 0.002, 0.00025, and 0.0002 and decay with a cosine annealing schedule. We train 300 epochs on both datasets and adopt the same training strategies for fully fine-tuning and linear fine-tuning. In inference, we obtain final predictions by averaging scores from 10 uniformly sampled temporal clips.

2.3. Video Retrieval

For nearest-neighbor video retrieval, we first use the pre-trained models without fine-tuning to extract features for both the training set and testing set. Each video will obtain 10 feature vectors by 10 uniformly sampled video clips. Then we average these features for each video and perform *L2* normalization on averaged features. Finally, for each testing video, we calculate its cosine similarities with all training videos. The evaluation metric is Recall at k (R@k), *i.e.*, a correct retrieval refers to that the top k nearest neighbours contain the correct class.

2.4. Temporal Action Localization

Temporal Action Localization(TAL) aims to generate temporal proposals which contain the action instances. Two metrics are used to evaluate the generated proposals: AR and AUC. The former is Average Recall rate with different tIoU thresholds, and the AUC is calculated by the area under the AR vs. Average Number of proposals (AN) curve, and the AN is varied from 0 to 100. For each video, we directly adopt the pre-trained models to extract 100 temporally uniform features as the input of BMN [7]. The optimizer for BMN is Adamw [8], with a learning rate of 0.001 and a weight decay of 1e-6. The learning rate decays with a cosine annealing schedule. We train BMN for 10 epochs and set the training batch size to 128.

3. Additional Experimental Results

3.1. Different temporal distance for VCL

We propose a simple δ_{max} to constrain the maximum distance between two sampled positive clips for visual consistency learning. To qualify the impact of δ_{max}, we evaluate different δ_{max} based on standard contrastive learning framework, the performance curves on action recognition task are shown in Figure A1 (a) and (b). We can observe that increasing δ_{max} may

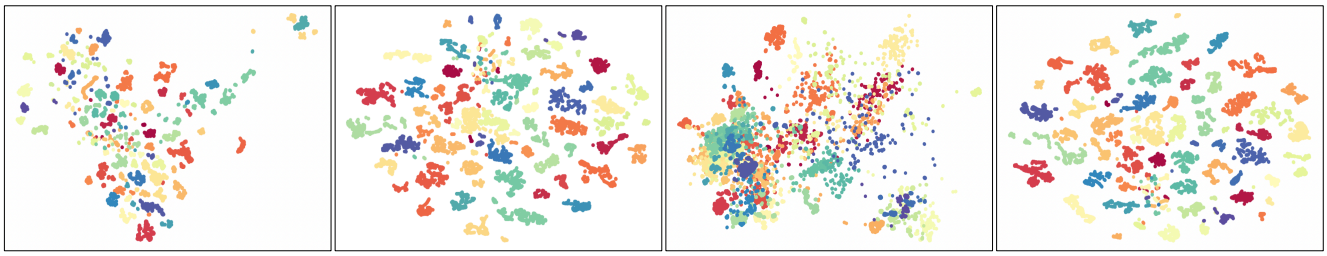

Figure A3. tSNE projection of video features in ActivieyNet-v1.3 dataset. Each color represents an untrimmed video. We present different datasets, *i.e.*, trimmed and untrimmed datasets, with standard contrastive learning and HiCo for pre-training.

hurt both fully fine-tuning and linear fine-tuning accuracy. However, the peak of linear fine-tuning appears at $\delta_{\text{max}} = 1s$ on both datasets. One possible reason is that forcing two semantic unrelated long-range clips to share the same feature embedding will confuse the network. Conversely, for two almost identical clips, the network can find shortcuts easily between them and fails to learn powerful representations.

3.2. Dataset Scales

We randomly select trimmed videos from the K400 dataset and find their untrimmed versions to generate multiple mini datasets with different scales but the same source. The S3D-G is pre-trained with HiCo on these datasets with the same training iteration. Figure A2 shows the linear evaluation for the learned representations on both HMDB51 and UCF101. We observe that HiCo consistently learns more powerful representations from untrimmed videos. This demonstrates that our HiCo can be generalized to any untrimmed dataset scale.

3.3. Visualization

In Figure A3, we explore the spatial-temporal representations learned by standard contrastive learning and HiCo on ActivieyNet-v1.3 dataset, using the tool of tSNE projection [12]. The S3D-G network is adopted as a feature extractor, and each color in the figure represents an untrimmed video. When pre-training with the standard contrastive learning framework, the separability of the features learned from untrimmed K400 is significantly worse than that from trimmed K400. This implies forcing different video clips with low visual similarity to share the same feature embedding seriously confuses the network. In comparison, our HiCo can always learn more robust representations regardless of the trimmed or untrimmed dataset.

References

[1] Dimitri P Bertsekas and John N Tsitsiklis. Neuro-dynamic programming: an overview. In *Proceedings of 1995 34th IEEE Conference on Decision and Control*, volume 1, pages 560–564. IEEE, 1995. 3

[2] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018. 3

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607. PMLR, 2020. 5

[4] Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013. 3

[5] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *CVPR*, pages 6546–6555, 2018. 6

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[7] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3889–3898, 2019. 6

[8] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[9] Yurii Nesterov. *Introductory lectures on convex optimization : a basic course*. Applied optimization. Kluwer Academic Publ., 2004. 2

[10] Boris Teodorovich Polyak. Gradient methods for minimizing functionals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 3(4):643–653, 1963. 2

[11] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, pages 6450–6459, 2018. 6

[12] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 7

[13] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999. 1

[14] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, pages 305–321, 2018. 6

[15] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017. 6