GLOBAL PROTOTYPE ENCODING FOR INCREMENTAL VIDEO HIGHLIGHTS DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Video highlights detection (VHD) is an active research field in computer vision, aiming to locate the most user-appealing clips given raw video inputs. However, most VHD methods are based on the closed world assumption, *i.e.*, a fixed number of highlight categories is defined in advance and all training data are available beforehand. Consequently, existing methods have poor scalability with respect to increasing highlight domains and training data. To address above issues, we propose a novel video highlights detection method named **G**lobal **P**rototype **E**ncoding (GPE) to learn incrementally for adapting to new domains via parameterized prototypes. To facilitate this new research direction, we collect a finely annotated dataset termed *LiveFood*, including over 5,100 live gourmet videos that consist of four domains: *cooking*, *eating*, *ingredients* and *presentation*. To the best of our knowledge, this is the first work to explore video highlights detection in the incremental learning setting, opening up new land to apply VHD for practical scenarios where both the concerned highlight domains and training data increase over time. We demonstrate the effectiveness of GPE through extensive experiments. Notably, GPE surpasses popular domain-incremental learning methods on *LiveFood*, achieving significant mAP improvements on all domains. The code and dataset will be made publicly available.

1 INTRODUCTION

The popularization of portable devices with cameras greatly promotes the creation and broadcasting of online videos. These sufficient video data serve as essential prerequisites for relevant researches, *e.g.* video summarization (Potapov et al., 2014; Song et al., 2015; Zhang et al., 2018; Fajtl et al., 2018; Zhu et al., 2021), video highlights detection (VHD) (Yang et al., 2015; Xiong et al., 2019; Lei et al., 2021; Bhattacharya et al., 2021), and moment localization (Liu et al., 2018; Zhang et al., 2020; Rodriguez et al., 2020), to name a few. Currently, most VHD methods are developed under the closed world assumption, which requires both the number of highlight domains and the size of training data to be fixed in advance. However, as stated in Rebuffi et al. (2017), natural vision systems are inherently incremental by consistently receiving new data from different domains or categories. Taking the gourmet video as an example, in the beginning, one may be attracted by the clips of eating foods, but lately, he/she may raise new interests in cooking and want to checkout the detailed cooking steps in the same video. This indicates that the target set the model needs to handle is flexible in the open world. Under this practical setting, all existing VHD methods suffer from the scalability issue: they are unable to predict both the old and the newly added domains, unless they retrain models on the complete dataset. Since the training cost on videos is prohibitive, it is thus imperative to develop new methods to deal with the above incremental learning issues.

Broadly speaking, there exist two major obstacles that hinder the development of incremental VHD: a high-quality VHD dataset with domain annotations and strong models tailored for this task. Recall existing datasets that are widely used in VHD research, including SumMe (Gygli et al., 2014), TVSum (Song et al., 2015), Video2GIF (Gygli et al., 2016), PHD (Garcia del Molino & Gygli, 2018), and QVHighlights (Lei et al., 2021), all of them suffer from threefold drawbacks: (1) only the feature representations of video frames are accessible instead of the raw videos, thus restricting the application of more powerful end-to-end models; (2) most datasets only have a limited number of videos with short duration and coarse annotations, which are insufficient for training deep models; (3) none of them has the video highlight domain or category labels, thus can not be directly used in

Figure 1: The *LiveFood* dataset. The row from top to bottom illustrates examples of vanilla clips, *ingredients* and *presentation* respectively. More samples are attached in Appendix A.2.

incremental learning. In order to bridge the gap between VHD and incremental learning, we first collect a high-quality gourmet dataset from live videos, namely *LiveFood*. It contains over 5,100 carefully selected videos with 197 hours in total. Four domains are finely annotated, *i.e., cooking*, *eating*, *ingredients* and *presentation*. These related but distinctive domains provide a new test bed for incremental VHD tasks.

To solve this new task, we propose a competitive model: **G**lobal **P**rototype **E**ncoding (GPE) to learn new highlight concepts incrementally while still retaining knowledge learned in previous video domains/data. Specifically, GPE first extracts frame-wise features using a CNN, then employs a transformer encoder to aggregate the temporal context to each frame feature, obtaining temporal-aware representations. Furthermore, each frame is classified by two groups of learnable prototypes: highlight prototypes and vanilla prototypes. With these prototypes, GPE optimizes a distance-based classification loss under L_2 metric and encourages incremental learning by confining the learned prototypes in new domains to be close to that previously observed. We systematically compare the GPE with different incremental learning methods on *LiveFood*. Experimental results show that GPE outperforms other methods on highlight detection accuracy (mAP) with much better training efficiency, using no complex exemplar selection or complicated replay schemes, strongly evidencing the effectiveness of GPE.

The main contributions of this paper are summarized as follows:

- We introduce a new task named incremental video highlights detection, which has important applications in practical scenarios. A high-quality *LiveFood* dataset is collected to facilitate research in this direction. *LiveFood* comprises over 5,100 carefully selected gourmet videos in high resolution, providing a new test bed for video highlights detection and domain-incremental learning tasks.
- We propose a novel end-to-end model for solving incremental VHD, *i.e.,* **G**lobal **P**rototype **E**ncoding (GPE). GPE can incrementally identify highlight and vanilla frames in new highlight domains via learning extensible and parameterized highlight/vanilla prototypes. GPE achieves superior performance compared with other incremental learning methods, improving the detection performance (mAP) by 1.57% on average. The above results suggest that GPE can serve as a strong baseline for future research.
- We provide comprehensive analyses of *LiveFood* as well as the proposed GPE model for deepening the understanding of both, as well as giving helpful insight for future development. We hope our work can inspire more researchers to work in incremental VHD, finally pushing forward the application of VHD in practical scenarios.

2 Related Work

Video Highlights Detection (VHD) is an important task in video-related problems. This line of research can be roughly divided into two groups, namely the ranking-based and regression-based

methods. Yao et al. (2016) employs a ranking model to learn the relationship between highlights and non-highlights, assigning higher scores to the positive clips. Saquil et al. (2021) utilizes multiple pairwise rankers to capture both the local and global information. Badamdorj et al. (2021) assigns higher scores to annotated clips based on dual-modals, *i.e.*, the visual and audio streams. Based on ranking methods, a lot of works aim to mitigate the expensive cost of human annotation using unsupervised techniques or priors, such as (Xiong et al., 2019; Badamdorj et al., 2022). Different from the above methods, regression-based methods predict the locations of highlights directly. Zhu et al. (2021) presents anchor-based and anchor-free approaches to predict the start and end timestamps, as well as the confidence score of highlights, therefore avoiding the laborious manual-designed post-processing. Moment-DETR (Lei et al., 2021) employs a transformer decoder to obtain the timestamps of specific clips based on different queries. Although the existing methods mentioned above boost the performance of VHD tasks, they substantially neglect the requirements of incremental learning in VHD, which is critical to the practical applications of VHD. In reality, numerous videos and new interests are created rapidly, thus demanding the VHD model to be capable of efficiently handling increasing highlight domains and data.

Incremental Learning (IL) is of great concern since the natural vision systems are inherently incremental. The main problem to solve in incremental learning is catastrophic forgetting, manifested as the forgetting of old classes or domains when learning new concepts. As investigated in Lange et al. (2022), the primary efforts deal with this issue in three aspects: using a memory buffer to store the representative data (Rebuffi et al., 2017; Isele & Cosgun, 2018; Rolnick et al., 2019; Yan et al., 2021; Lange & Tuytelaars, 2021), adopting regularization terms to constrain the change of model's weights or outputted logits (Kirkpatrick et al., 2016; Zenke et al., 2017; Schwarz et al., 2018) and performing parameter isolation to dedicate different model parameters for each task (Fernando et al., 2017; Mallya & Lazebnik, 2018; Rosenfeld & Tsotsos, 2020). The memory buffer replays previous samples while learning new concepts to eliminate forgetting, however, it may lead to overfitting on the stored sub-set and incur heavy memory costs. The regularization-based methods penalize the model if some characteristics are changed during the next training stage, resulting in the domination of the so-called essential characters. Parameter isolation grows new branches for new tasks, raising prohibitive colossal architectures. To mitigate the negative effects of existing methods, GPE employs prototype learning combined with distance measurement to perform binary classification (highlight vs. vanilla frames). Prototypes are essentially the most representative features learned across the whole training data, circumventing both the issue of overfitting on the sub-set and the unbearable costs of storing raw data. Besides, we constrain the change of prototypes between stages which is an overall refinement instead of minority domination. The prototypes from previous stages are inherited during training in the current domains, so as to maintain global consistency while leaving room for adjustment and improvements.

3 Problem Statement

In incremental VHD, the training procedure consists of several consequent tasks built on disjoint datasets with distribution shifts. Assuming that we have T tasks in total, and this yields a training data stream $\{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_T\}$ where $\mathcal{T}_i \cap \mathcal{T}_j = \varnothing$ if $i \neq j$. Each training task $\mathcal{T}_t$ is represented as $\{(x_i^t, y_i^t)\}_{i=1}^{n_t}$ where $x_i^t \in \mathcal{X}$ denotes the whole frame set of *i-th* training video in stage t, y_i^t is its corresponding frame-wise label (*i.e.*, a binary vector indicating highlight/vanilla frames), and n_t represents the number of accessible training data pairs. Moreover, we use $\{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_T\}$ to describe the corresponding domains included in training tasks $\{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_T\}$. Note $\mathcal{D}_i \neq \mathcal{D}_j$ if $i \neq j$. Specifically, in our proposed $LiveFood$, we split the training videos into four disjoint subset $\{\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3, \mathcal{T}_4\}$, and the corresponding domains are denoted as $\{\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3, \mathcal{D}_4\}$. Considering that a video may consist of more than one domain, we further constrain that the domains appearing in $\mathcal{D}_{t-1}$ is a sub-collection of that in $\mathcal{D}_t$. Formally, let $\mathcal{S}_t$ denotes all possible combinations of domains presented in $\mathcal{D}_t$, and $\mathcal{C}_t$ is the domain combinations appearing in videos of $\mathcal{T}_t$, we have

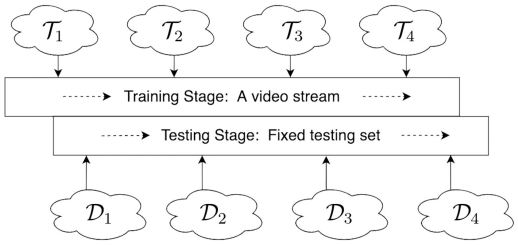
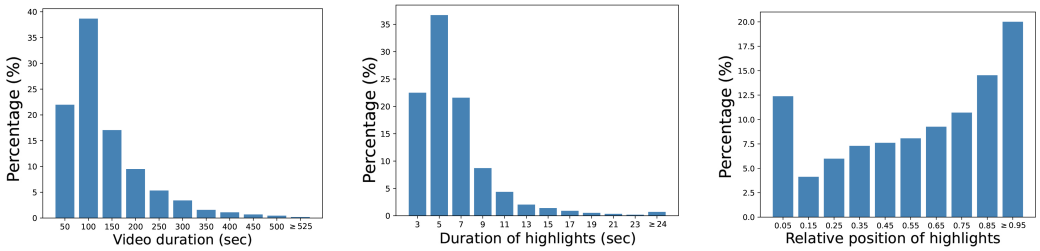


Figure 2: Illustration of conducting incremental VHD on *LiveFood*.

$\mathcal{C}_1 = \mathcal{S}_1$ and $\mathcal{C}_t = \mathcal{S}_t \setminus \bigcup_{i=1}^{t-1} \mathcal{S}_i$. More concretely, let d_i denotes a specific domain label, then if $\mathcal{D}_1 = \{d_1\}$, $\mathcal{D}_2 = \{d_1, d_2\}$, and $\mathcal{D}_3 = \{d_1, d_2, d_3\}$, we have $\mathcal{C}_1 = \{d_1\}$, $\mathcal{C}_2 = \{d_2, (d_1, d_2)\}$ and $\mathcal{C}_3 = \{d_3, (d_1, d_3), (d_2, d_3), (d_1, d_2, d_3)\}$. In above example, $\mathcal{C}_2 = \{d_2, (d_1, d_2)\}$ indicates that the videos in $\mathcal{T}_2$ can contain the domain of d_2 or the mixture of d_1 and d_2. Videos that merely contain the domain of d_1 are excluded from $\mathcal{T}_2$ since the intention of incremental VHD is to effectively learn new concepts while remembering what are already learned in past data. The testing set contains mixed videos including **all domains**, and during task $\mathcal{T}_t$, only the domain **appearing in** $\mathcal{D}_t$ is treated as positive when evaluating the performance.

4 THE LIVEFOOD DATASET

Video Selection. We collect online gourmet videos with high resolution. As introduced in Xiong et al. (2019), shorter videos are more likely to contain attractive clips and thus have more hits, while longer videos are usually boring. Taking this prior into consideration, we filter out both the extremely short (less than 30 seconds) which may contain insufficient gourmet content to learn from and long (over 15 minutes) videos which users generally pay less attention to. After that, all reserved raw videos are viewed by qualified workers to check whether the content of videos is gourmet-related or not, eliminating the effects of incorrect category annotation. Only videos that pass the aforementioned checks are selected for subsequent annotation tasks, in order to guarantee the quality of *LiveFood*. Figure 3 (a) shows the duration distribution of the videos across all domains.



(a) (b) (c)

Figure 3: Statistical results of the proposed *LiveFood*. (a) shows the distribution of video duration. (b) illustrates the distribution of highlight durations. Most highlight clips are shorter than 15 seconds. (c) shows the relative position of each attractive clip w.r.t. corresponding videos. The highlights distribute evenly across whole videos, evidencing the good diversity of *LiveFood*.

Highlights Annotation. We define four highlight domains that are generally presented in collected videos, namely *cooking*, *eating*, *ingredients* and *presentation*. For each domain, a video clip is accepted as a satisfactory highlight if it meets the following criteria in Table 1.

Table 1: Basic description of annotated domains in *LiveFood*.

Domain	Description
cooking	The process demonstrates the chef's exquisite cooking skills.
eating	People enjoy foods with exaggerated or satisfied expressions on their faces.
ingredients	The video clip shows at least three kinds of rare ingredients in high quality.
presentation	A well-displayed meal to make it looks more appetizing.

The annotators are required to glance over the whole video first to locate the coarse position of attractive clips. Afterward, the video is annotated at frame level from the candidate position to verify the exact start and end timestamps of highlights. Meanwhile, we introduce a strict double-check mechanism (cf. Appendix A.3) to further guarantee the quality of annotations. Since selecting the timestamps of highlights is partly subjective, both objective and subjective verifications are necessary in quality control. Concerning the consistent visual feeling of videos, we restrict the highlights to be longer than three seconds, and less than two minutes to avoid being tedious.

Data Statistics. Figure 3 depicts the statistical results of our proposed *LiveFood*, including the distribution of video duration, the distribution of highlights duration, and the relative position of

center timestamp with respect to each video, etc. Besides, in Table 2, we also compare the proposed *LiveFood* with existing VHD datasets such as SumMe (Gygli et al., 2014), YouTubeHighlights (Sun et al., 2014), Video2GIF (Gygli et al., 2016), PHD (Garcia del Molino & Gygli, 2018) and QVHighlights (Lei et al., 2021) for better illustrating the differences among them.

Table 2: Comparison between the proposed *LiveFood* and existing datasets.

Dataset	Year	Contents	Label domain / class	Total number of videos / highlights	Avg. len. (sec) of videos / highlights
SumMe	2014	Open	✗	25 / 390	120.0 / -
YouTubeHighlights	2014	Activity	✗	712 / -	143.0 / 2.0
Video2GIF	2016	Open	✗	80K / 98K	332.0 / 5.2
PHD	2018	Open	✗	119K / 228K	440.2 / 5.1
QVHighlights	2021	Vlog / News	✗	10.2K / 10.3K	150 / 24.6
LiveFood (ours)	2022	Gourmet	✓	5.1K / 14.3K	136.5 / 6.4

As demonstrated in Table 2, the SumMe and YouTubeHighlights only contain a small number of videos and annotations, which makes them insufficient for training deep models. The Video2GIF and PHD are edited by online users and lack strict quality control mechanisms. Thus the reliability of datasets may be undermined. The newly released QVHighlights can not be used for incremental learning since it does not have domain annotations. Besides, the average length of clips in QVHighlights is pretty long: nearly one-fifth of each video is annotated as attractive clips, thus causing the selected clips less discriminative compared to the vanilla clips. Different from the above datasets, *LiveFood* provides gourmet videos with finely annotated domain labels, making it suitable for domain-incremental VHD tasks.

5 METHOD

GPE aims to tackle forgetting while still improving by learning new concepts. As analyzed in Section 2, conventional incremental learning methods have shortcomings, such as overfitting on replayed data, limited flexibility, and unbearable growing architectures. Distinguished from them, GPE employs prototypes together with distance measurement to solve the classification problem. Prototypes are compact and concentrated features learned on the training data, mitigating the effects of overfitting on the stored sub-set. In addition, by using global and dynamic prototypes, we endow the model with appealing capability for further refinement when fed with new data (M_{f} in Figure 4) or accommodation to new domain concepts (M_{d} in Figure 4).

Architecture. Inspired by Carion et al. (2020), GPE employs the combination of convolution- and attention-based models to extract features. Concretely, a ConvNeXt (Liu et al., 2022) pre-trained on ImageNet (Russakovsky et al., 2015) is used to extract spatial features of input video frames. After that, a transformer encoder with multi-heads is used to perform temporal fusion, generating global representations based on the whole video frames. With the transformation of a feedforward network (FFN) which consists of fully-connected layers, each frame is classified based on the distance to learnable prototypes. We aim to learn two groups of trainable prototypes with the same shape, namely the highlight (positive) and vanilla (negative) prototypes. By denoting the dimensionality of output feature of transformer as m and the number of prototypes within each group as k, both the highlight and vanilla prototypes can be represented as a matrix with shape $k \times m$. By utilizing L_2 distance as the distance measurement between each feature and prototype, we obtain the pair-wise distance between features and each group of prototypes. Formally, we use h, H, and V to represent the transformer feature, the highlight prototypes and the vanilla prototypes. $g_\phi(\cdot)$ denotes the FFN module. $d(\cdot)$ is the L_2 distance between a feature-prototype pair. The distance from feature h to H and V are formulated as:

$$d_H = \min_{i=1:k} d(g_\phi(h), H_i), \quad d_V = \min_{i=1:k} d(g_\phi(h), V_i), \tag{1}$$

where the subscript i represents the i-th prototype. The distance is mapped to probability P_H using the softmax function which can be understood as the confidence of assigning feature h to highlights.

$$P_H = \frac{\exp(-d_H)}{\exp(-d_H) + \exp(-d_V)}. \tag{2}$$

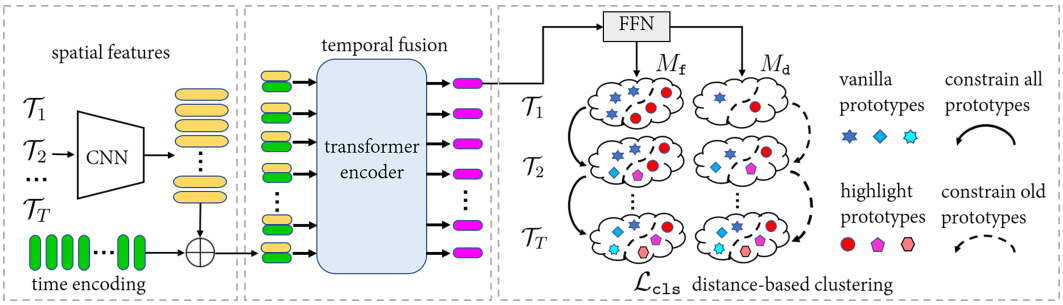


Figure 4: The proposed GPE framework. $\{\mathcal{T}_i\}_{i=1}^{T}$ indicates a training stream with T tasks. In the right-most part, $M_{\mathtt{f}}$ and $M_{\mathtt{d}}$ represents the fixed and dynamic modes of GPE. $M_{\mathtt{f}}$ defines the number of prototypes in advance and refines them across different stages. A restriction on the magnitude of change amplitude is imposed during learning (cf. Eq. 5). $M_{\mathtt{d}}$ dynamically adds new prototypes into the learning process when dealing with new domains. The change restriction is only applied on inherited prototypes. This mode is more suitable when learning on a large amount of domains. Each prototype in above figure is equivalent to a row of V or H (cf. Eq. 1).

Then, we use cross-entropy loss to optimize the model through gradient back-propagation:

$$\mathcal{L}_{\mathtt{cls}} = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(P_H) + (1-y_i)\cdot\log(1-P_H) \tag{3}$$

where N represents the size of training frames and y_i equals 1 if the i-th frame is annotated as highlights otherwise 0.

Learning with incremental domains. We detail the learning of $M_{\mathtt{f}}$ (Figure 4) in this section, and the dynamic mode $M_{\mathtt{d}}$ can be easily derived by only restricting the change of inherited prototypes and training newly added prototypes freely. We use $h_\theta(\cdot)$ parameterized by θ to indicate the feature extractor jointly constructed by a ConvNeXt and a transformer encoder. The FFN $g_\phi(\cdot)$ is parameterized by ϕ. Recall the outputted feature is h. Both the vanilla and highlight prototypes are denoted by π for simplicity. With the help of these notations, the classification loss built in Eq. 3 is abbreviated as $\mathcal{L}_{\mathtt{cls}}(\theta,\phi,\pi)$. We expand the definition of distance measurement for evaluating the distance between given two prototypes. For two sets of learned prototypes $\pi^{(t)}$ and $\pi^{(t+1)}$, the distance between them is calculated as:

$$d(\pi^{(t)},\pi^{(t+1)}) = \frac{1}{k}\sum_{i=1}^{k}\sqrt{\sum_{j=1}^{m}(\pi_{i,j}^{(t)}-\pi_{i,j}^{(t+1)})^2} \tag{4}$$

During the training phase T, the model inherits trained prototypes $\pi^{(T-1)}$ from the former stage. For the incremental need, we tackle the catastrophic forgetting issue by restricting the change of prototypes, guaranteeing the awareness that the model has learned towards the observed domains. With the above formulations, we consider the following constrained nonlinear optimization problems:

$$\begin{aligned} \min_{\theta,\phi,\pi} \quad & \mathcal{L}_{\mathtt{cls}}(\theta,\phi,\pi) \\ \mathtt{s.t.} \quad & d(\pi^{(T-1)},\pi) \le \gamma \end{aligned} \tag{5}$$

where γ is the tolerable change introduced to the observed prototypes. The optimal result that meets the above restriction is $(\theta^{(T)},\phi^{(T)},\pi^{(T)})$. Instead of solving the complex nonlinear problem, we resort to its corresponding empirical dual formulation. With an auxiliary positive Lagrange multiplier λ, the optimization objective in Eq. 5 is transformed into the following manner:

$$\begin{aligned} S^{(T)} &= (\theta^{(T)},\phi^{(T)},\pi^{(T)}) \\ &= \max_{\lambda}\min_{\theta,\phi,\pi} \; L(\theta,\phi,\pi,\lambda) \\ &= \max_{\lambda}\min_{\theta,\phi,\pi} \; \mathcal{L}_{\mathtt{cls}}(\theta,\phi,\pi) + \lambda[d(\pi^{(T-1)},\pi)-\gamma] \end{aligned} \tag{6}$$

where $S^{(T)}$ indicates the optimal solution in training stage T. We update the trainable parameters (*i.e.*, θ, ϕ, and π) and the empirical Lagrangian variable λ alternatively and iteratively:

$$\begin{aligned}
\theta &\leftarrow \theta - \eta \frac{\partial \mathcal{L}_{\texttt{cls}}(\theta, \phi, \pi)}{\partial \theta} \\
\phi &\leftarrow \phi - \eta \frac{\partial \mathcal{L}_{\texttt{cls}}(\theta, \phi, \pi)}{\partial \phi} \\
\pi &\leftarrow \pi - \eta \frac{\partial L(\theta, \phi, \pi, \lambda)}{\partial \pi} \\
\lambda &\leftarrow \max\{\lambda + \eta[d(\pi^{(T-1)}, \pi) - \gamma], 0\}
\end{aligned} \tag{7}$$

where η is the learning rate of trainable parameters and λ is the multiplier in the dual step. With the above analysis, the incremental training and inference pipeline is summarized in Algorithm 1.

Algorithm 1: Global Prototype Encoding for Incremental Video Highlights Detection.

Input: training data $\mathcal{X}$ with k stages; initial $h_\theta(\cdot)$, $g_\phi(\cdot)$, and π; initial η, γ, and λ.
Output: trained $h_\theta(\cdot)$, $g_\phi()\cdot$, and π at each stage.

```
while Training do
    for stage i=1:k do
        Reset η, γ, and λ;
        repeat
            Sample a batch data {x, y} from X within stage i;
            Calculate outputted logits: g_φ(h_θ(x));
            Calculate the loss function with Eq 6;
            Update θ, φ, π, and λ with Eq 7;
        until θ, φ, and π are converged;
while Inference do
    for stage i=1:k do
        repeat
            Sample {x, y} from the testing set at stage i;
            Calculate logits: l = g_φ(h_θ(x));
            Classify inputs based on the distance between l and π;
        until the testing set is ∅;
        Average the mAP cross domains.
    Average the mAP cross all stages.
```

6 EXPERIMENT

We introduce the details of the evaluation protocol and experimental results in this section.

6.1 EXPERIMENTAL SETUP

Data and Evaluation Protocol. *LiveFood* contains 4928 videos for training and 261 videos for testing. We randomly split 15% of the 4928 videos for validation. $\mathcal{T}_1$, $\mathcal{T}_2$, $\mathcal{T}_3$ and $\mathcal{T}_4$ consist of 3380, 854, 393, and 113 videos respectively. $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$ and $\mathcal{D}_4$ are {*presentation*}, {*presentation*, *eating*}, {*presentation*, *eating*, *ingredients*}, and {*presentation*, *eating*, *ingredients*, *cooking*}. We report the mAP on testing set following previous works (Yao et al., 2016; Xiong et al., 2019).

GPE. GPE only finetunes the last layer of ConvNeXt during training. The transformer encoder has 8 heads and 3 layers. The feedforward module $g_\phi(\cdot)$ is a multi-layer perception with 3 linear layers activated by ReLU. Both the vanilla and highlight prototypes are formulated as 40 vectors with dimensions of 128. During every stage, the model is trained with 300 epochs, and the learning rate halves every 70 epochs starting with 1e-3. GPE is initialized randomly during the first stage. In later stages, it trains prototypes in $M_{\texttt{f}}$ by starting from weights learned in the former stage. In $M_{\texttt{d}}$, all prototypes learned in previous stages are inherited and trained as similar as in $M_{\texttt{d}}$. In addition to these inherited prototypes, in each stage, new prototypes are randomly initialized and added to $M_{\texttt{d}}$ so as to enhance the learning of new concepts.

Regularization-based methods. SI (Zenke et al., 2017) and oEWC (Schwarz et al., 2018) are representatives. These schemes update the model with a compromised loss function, utilizing both the overall importance of previous stages and the current task to eliminate forgetting.

Replay methods. ER (Rolnick et al., 2019) and DER (Buzzega et al., 2020) use a memory buffer to store the representative data from the previous task to defend against forgetting. The buffer size is set to 200 if not specified.

Figure 5: The observed highlights across different training stages.

Lower bound (Lb). In each stage, GPE is trained without constraints (cf. Eq. 5), suffering from severe catastrophic forgetting and rendering performance drops with the increasing of tasks.

Upper bound (Ub). In each stage, GPE is trained with data from all stages. Thus it is free from the forgetting issue and provides the upper bound performance for all incremental learning methods.

6.2 Main Result

Comparison with existing IL methods. The experimental results are depicted in Table 3. We highlight the upper bound results with gray background. It can be observed from Table 3 that the vanilla GPE ($M_{\mathtt{f}}$) surpasses the lower bound with an improvement of 2.91% mAP. Compared to the classic IL methods SI and oEWC, GPE outperforms them by a remarkable margin, yielding at least 1.57% performance gain on mAP. Moreover, when equipped with the same replay schemes as Yan et al. (2021), GPE achieves 1.06% and 2.71% mAP gain compared to DER and ER. The above results clearly demonstrate the effectiveness of GPE in tackling the incremental VHD task.

Table 3: Comparison of GPE with existing incremental learning methods on *LiveFood*. We evaluate their frame-wise mAP performance.

mAP	Lb	SI	oEWC	ER*	DER*	Ub	GPE($M_{\mathtt{f}}$)	GPE($M_{\mathtt{d}}$)	GPE*
$\mathcal{T}_1$	36.13	36.16	36.13	35.79	36.17	36.15	36.14	36.21	36.17
$\mathcal{T}_2$	30.86	31.84	31.82	31.38	33.14	37.38	35.82	36.13	36.62
$\mathcal{T}_3$	29.18	30.72	30.51	29.13	32.52	36.90	31.87	32.74	33.15
$\mathcal{T}_4$	25.89	28.73	28.67	29.06	30.11	36.30	29.88	30.15	30.27
Avg.	30.52	31.86	31.78	31.34	32.99	36.68	<u>33.43</u>	<u>33.90</u>	**34.05**

*: using memory buffer to replay samples.

Visualization of highlight scores across training stages. We investigate the effects of observed prototypes across different training stages. In Figure 5, we present the highlight detection results of GPE in the first and the last training stage. For comparison, we also provide the prediction of DER. In the curves shown in Figure 5, the blue and orange points indicate the highlight scores of each frame predicted by GPE (*i.e.*, P_H in Eq. 2) during the first task $\mathcal{T}_1$ and the final task $\mathcal{T}_4$. The green points represent the predicted scores of DER in $\mathcal{T}_4$. It is clear that GPE can learn new concepts, *e.g.*, *cooking* while keeping the memory of *presentation* learned in the first stage. This result is in line with our motivation that a strong incremental VHD model should be able to cover both the past and new concepts. In contrast, since DER employs stored data to strengthen memory, it still has the drawback of being prone to forgetting due to the limited buffer size. This is demonstrated by its assigning much lower scores to the old domain of *presentation* when learning *cooking*.

Scalability of GPE in the dynamic manner. We investigate the generalization ability of dynamic GPE ($M_{\mathtt{d}}$) in the scenario where the model needs to handle a large number of domains. We consider the R-MNIST dataset containing a series of rotated digits with different degrees between $[0, \pi)$, where each degree represents a domain. For a fair comparison, we use the identical settings as Buzzega et al. (2020), yielding a stream with 20 subsequent tasks. Note that no augmentation techniques are used. In this experiment, GPE is simplified to be a small network with 2 fully-connected layers followed by ReLU. The number of prototypes is set to 5 per class. Therefore each task $\mathcal{T}_t$ has $5t$ prototypes per class by inheriting from previous stages and generating 5 new prototypes for each class. The old prototypes are forbidden to change too much. λ and γ are 10 and 1e-2. All other experimental settings follow Buzzega et al. (2020).

Table 4: Average classification accuracy on R-MNIST.

Method	SI	oEWC	ER	GEM	FDR	GSS	HAL	DER	GPE (M_{d})	GPE (M_{d})
Buffer Size	✗	✗	200	200	200	200	200	200	✗	200
Avg Acc.	71.91	77.35	85.01	80.80	85.22	79.50	84.02	90.04	$85.42_{\pm 0.13}$	$\mathbf{90.17}_{\pm 0.25}$

Results depicted in Table 4 demonstrate that the dynamic GPE surpasses most conventional methods, including the regularization-based and replay methods. Notably, the dynamic GPE achieves 85.42% average accuracy across 20 tasks, comparable to ER (Rolnick et al., 2019) and FDR (Benjamin et al., 2019) while outperforming other methods with a considerable margin. We further adopt the identical replay schemes as DER (Yan et al., 2021), and this helps the dynamic GPE achieve 90.17% top-1 accuracy, which is established as a new comparable *state-of-the-art* approach.

6.3 Ablation Study

Ablation on the initial number of prototypes k. The number of prototypes reflects the model's capacity. Too many prototypes increase the training cost while too few lead to underfitting. Results shown in Table 5 provide a comparison between the training cost and performance under the fixed mode of GPE (M_{f}) on *LiveFood*. It is observed that with the increasing initial quantity of prototypes k, the average mAP over all tasks consistently increases. However, by comparing the last two rows in Table 5, we notice that the performance gain between $k = 40$ and $k = 50$ is marginal though more parameters are introduced. Therefore, we set k to 40 throughout experiments to strike a good balance between accuracy and efficiency.

Table 5: Ablations on the initial number of prototypes k.

k	mAP ($\gamma = 5.0$)				
	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$	$\mathcal{T}_4$	Avg.
10	35.31	33.48	29.18	25.98	30.99
20	35.82	34.66	30.72	28.73	32.35
30	36.13	35.70	30.51	28.67	32.75
40	36.14	35.82	31.87	29.88	33.43
50	36.21	35.88	31.90	29.94	**33.48**

Table 6: Ablations on the changing constraints of distance γ.

γ	mAP ($k = 40$)				
	$\mathcal{T}_1$	$\mathcal{T}_2$	$\mathcal{T}_3$	$\mathcal{T}_4$	Avg.
1e-3	36.14	34.26	30.17	26.43	31.75
1.0	36.13	34.97	30.62	27.41	32.28
3.0	36.15	35.50	31.44	28.27	32.84
5.0	36.14	35.82	31.87	29.88	**33.43**
15.0	36.14	34.92	30.80	28.12	32.50

Ablation on distance constraint γ. Extremely small γ hinders the model from learning new concepts since the prototypes are almost unchanged. In contrast, too large γ may lead to catastrophic forgetting since the model may heavily overfit to the newly observed data. As shown in Table 6, we set k to 40 by default and investigate the effects of γ with different values. In our experiments, we find the distance between the vanilla and highlight prototypes after $\mathcal{T}_1$ is less than 15, so it is set as the upper bound of γ. From Table 6, we can see that when γ is small, say 1e-3, GPE can hardly learn the new contents, only achieving similar mAP compared to the lower bound method as shown in Table 3. By enlarging γ, the average mAP increases consistently from 31.75 to 33.43. When γ is 15, the model suffers from the forgetting issue as explained before, resulting in a near 1.0% performance drop. Consequently, we set k and γ to 40 and 5 by default.

7 Conclusion

In this paper, we introduce a new task: incremental video highlight detection, aiming to perform VHD in the practical scenario where both the highlight domains and data increase over time. To pave the road in this new direction, we collect a high-quality video gourmet dataset *LiveFood* which contains four fine-annotated domains. Then we propose a new end-to-end model named **G**lobal **P**rototype **E**ncoding (GPE) to learn incrementally to adapt to new highlight domains. Extensive experimental results clearly demonstrate the effectiveness of our method. We hope this work serves to inspire other researchers to work on this new and critical task.

REFERENCES

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 11816–11825, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/e562cd9c0768d5464b64cf61da7fc6bb-Abstract.html.

Taivanbat Badamdorj, Mrigank Rochan, Yang Wang, and Li Cheng. Joint visual and audio learning for video highlight detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8127–8137, October 2021.

Taivanbat Badamdorj, Mrigank Rochan, Yang Wang, and Li Cheng. Contrastive learning for unsupervised video highlight detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14042–14052, June 2022.

Ari S. Benjamin, David Rolnick, and Konrad P. Körding. Measuring and regularizing networks in function space. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=SkMwpiR9Y7.

Uttaran Bhattacharya, Gang Wu, Stefano Petrangeli, Viswanathan Swaminathan, and Dinesh Manocha. Highlightme: Detecting highlights from human-centric videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8157–8167, October 2021.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/b704ea2c39778f07c617f6b7ce480e9e-Abstract.html.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I*, volume 12346 of *Lecture Notes in Computer Science*, pp. 213–229. Springer, 2020. doi: 10.1007/978-3-030-58452-8_13. URL https://doi.org/10.1007/978-3-030-58452-8_13.

Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co^{2}l: Contrastive continual learning. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pp. 9496–9505. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00938. URL https://doi.org/10.1109/ICCV48922.2021.00938.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Hkf2_sC5FX.

Arslan Chaudhry, Albert Gordo, Puneet K. Dokania, Philip H. S. Torr, and David Lopez-Paz. Using hindsight to anchor past knowledge in continual learning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 6993–7001. AAAI Press, 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/16861.

Jiri Fajtl, Hajar Sadeghi Sokeh, Vasileios Argyriou, Dorothy Monekosso, and Paolo Remagnino. Summarizing videos with attention. In Gustavo Carneiro and Shaodi You (eds.), *Computer Vision - ACCV 2018 Workshops - 14th Asian Conference on Computer Vision, Perth, Australia,*

December 2-6, 2018, Revised Selected Papers, volume 11367 of *Lecture Notes in Computer Science*, pp. 39–54. Springer, 2018. doi: 10.1007/978-3-030-21074-8_4. URL https://doi.org/10.1007/978-3-030-21074-8_4.

Chrisantha Fernando, Dylan Banarse, Charles Blundell, Yori Zwols, David Ha, Andrei A. Rusu, Alexander Pritzel, and Daan Wierstra. Pathnet: Evolution channels gradient descent in super neural networks. *CoRR*, abs/1701.08734, 2017. URL http://arxiv.org/abs/1701.08734.

Ana Garcia del Molino and Michael Gygli. PHD-GIFs: Personalized Highlight Detection for Automatic GIF Creation. In *Proceedings of the 2018 ACM on Multimedia Conference*, MM '18, New York, NY, USA, 2018. ACM.

Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *ECCV*, 2014.

Michael Gygli, Yale Song, and Liangliang Cao. Video2gif: Automatic generation of animated gifs from video. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 1001–1009. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.114. URL https://doi.org/10.1109/CVPR.2016.114.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

David Isele and Akansel Cosgun. Selective experience replay for lifelong learning. In Sheila A. McIlraith and Kilian Q. Weinberger (eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 3302–3309. AAAI Press, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16054.

James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *CoRR*, abs/1612.00796, 2016. URL http://arxiv.org/abs/1612.00796.

Matthias De Lange and Tinne Tuytelaars. Continual prototype evolution: Learning online from non-stationary data streams. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pp. 8230–8239. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00814. URL https://doi.org/10.1109/ICCV48922.2021.00814.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory G. Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(7):3366–3385, 2022. doi: 10.1109/TPAMI.2021.3057446. URL https://doi.org/10.1109/TPAMI.2021.3057446.

Jie Lei, Tamara L Berg, and Mohit Bansal. Detecting moments and highlights in videos via natural language queries. *Advances in Neural Information Processing Systems*, 34:11846–11858, 2021.

Meng Liu, Xiang Wang, Liqiang Nie, Qi Tian, Baoquan Chen, and Tat-Seng Chua. Cross-modal moment localization in videos. In Susanne Boll, Kyoung Mu Lee, Jiebo Luo, Wenwu Zhu, Hyeran Byun, Chang Wen Chen, Rainer Lienhart, and Tao Mei (eds.), *2018 ACM Multimedia Conference on Multimedia Conference, MM 2018, Seoul, Republic of Korea, October 22-26, 2018*, pp. 843–851. ACM, 2018. doi: 10.1145/3240508.3240549. URL https://doi.org/10.1145/3240508.3240549.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 6467–6476, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/f87522788a2be2d171666752f97ddebb-Abstract.html.

Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 7765–7773. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00810. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Mallya_PackNet_Adding_Multiple_CVPR_2018_paper.html.

Danila Potapov, Matthijs Douze, Zaïd Harchaoui, and Cordelia Schmid. Category-specific video summarization. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI*, volume 8694 of *Lecture Notes in Computer Science*, pp. 540–555. Springer, 2014. doi: 10.1007/978-3-319-10599-4_35. URL https://doi.org/10.1007/978-3-319-10599-4_35.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 5533–5542. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.587. URL https://doi.org/10.1109/CVPR.2017.587.

Cristian Rodriguez, Edison Marrese-Taylor, Fatemeh Sadat Saleh, HONGDONG LI, and Stephen Gould. Proposal-free temporal moment localization of a natural-language query in video using guided attention. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P. Lillicrap, and Gregory Wayne. Experience replay for continual learning. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 348–358, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/fa7cdfad1a5aaf8370ebeda47a1ff1c3-Abstract.html.

Amir Rosenfeld and John K. Tsotsos. Incremental learning through deep adaptation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(3):651–663, 2020. doi: 10.1109/TPAMI.2018.2884462. URL https://doi.org/10.1109/TPAMI.2018.2884462.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Yassir Saquil, Da Chen, Yuan He, Chuan Li, and Yong-Liang Yang. Multiple pairwise ranking networks for personalized video summarization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1718–1727, October 2021.

Jonathan Schwarz, Wojciech M. Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. *ArXiv*, abs/1805.06370, 2018.

Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. Tvsum: Summarizing web videos using titles. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pp. 5179–5187. IEEE Computer Society, 2015. doi: 10.1109/CVPR.2015.7299154. URL https://doi.org/10.1109/CVPR.2015.7299154.

Min Sun, Ali Farhadi, and Steven M. Seitz. Ranking domain-specific highlights by analyzing edited videos. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, volume 8689 of *Lecture Notes in Computer Science*, pp. 787–802. Springer, 2014. doi: 10.1007/978-3-319-10590-1_51. URL https://doi.org/10.1007/978-3-319-10590-1_51.

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

Bo Xiong, Yannis Kalantidis, Deepti Ghadiyaram, and Kristen Grauman. Less is more: Learning highlight detection from video duration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3014–3023, June 2021.

Huan Yang, Baoyuan Wang, Stephen Lin, David P. Wipf, Minyi Guo, and Baining Guo. Unsupervised extraction of video highlights via robust recurrent auto-encoders. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pp. 4633–4641. IEEE Computer Society, 2015. doi: 10.1109/ICCV.2015.526. URL https://doi.org/10.1109/ICCV.2015.526.

Ting Yao, Tao Mei, and Yong Rui. Highlight detection with pairwise deep ranking for first-person video summarization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 982–990. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.112. URL https://doi.org/10.1109/CVPR.2016.112.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3987–3995. PMLR, 2017. URL http://proceedings.mlr.press/v70/zenke17a.html.

Ke Zhang, Kristen Grauman, and Fei Sha. Retrospective encoders for video summarization. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (eds.), *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VIII*, volume 11212 of *Lecture Notes in Computer Science*, pp. 391–408. Springer, 2018. doi: 10.1007/978-3-030-01237-3_24. URL https://doi.org/10.1007/978-3-030-01237-3_24.

Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. Learning 2d temporal adjacent networks for moment localization with natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 12870–12877, 2020.

Wencheng Zhu, Jiwen Lu, Jiahao Li, and Jie Zhou. Dsnet: A flexible detect-to-summarize network for video summarization. *IEEE Trans. Image Process.*, 30:948–962, 2021. doi: 10.1109/TIP.2020.3039886. URL https://doi.org/10.1109/TIP.2020.3039886.

A APPENDIX

A.1 ETHICS, DATA AND PRIVACY

We solemnly declare that the highest standard is strictly held to obey all regulations/rules regarding matters of ethics codes, data safety, and privacy protection. Below we introduce the main measures taken to eliminate any potential issues.

Data Desensitization. All videos appearing in *LiveFood* are collected from the internet and publicly available. When using these videos after their creators' authorization, we remove the private information related to video owners to protect their privacy. These pieces of information include but are not limited to identifiers, meta-data, addresses, and profiles.

Data Storage. We highly value the efforts the creators spent on their creativities. The video owners have the right to keep or delete their uploaded videos online. To protect the creators' legal rights and interests, we store **NO** videos locally. The *LiveFood* consists of video links and their corresponding annotations. Videos appearing in *LiveFood* become unavailable if their owners delete them, and we respect the creators' activities.

User Privacy. We significantly respect user privacy. All videos in *LiveFood* are authorized by their authors for non-commercial algorithm research, and we have informed the video owners of the purposes and potential effects of using these videos. Besides, the user identifiers, video timestamps, and other profiles that may reflect private information are removed to protect privacy.

Licence. *LiveFood* can only be used for non-commercial research (cf. Appendix A.7).

Annotator Related. The annotators are skilled and experienced in data preparation. Their efforts and time spent on selecting video highlights are highly praised. We emphasize that although this annotation task is labor-intensive, it is finished strictly following the working hours standard. Besides, we provide fair compensation and salary for their valuable giving. The whole annotation procedure takes a large amount of time and money.

A.2 VISUALIZATION OF VIDEO ANNOTATIONS

Our original purpose in collecting the *LiveFood* is to promote the research in incremental learning and video highlight detection. **We stress that *LiveFood* is strictly confined to academic research only**. In the main body of the manuscript, we do not present video samples containing faces due to preserve privacy. For helping grasp the main points of *LiveFood*, we show below a bunch of video samples along with their domain annotations. Note that for preserving the privacy of video creators, all faces appearing in the figures are blurred. The following six visualization samples, *i.e.*, `video_1/.../6` from Figure 6 to Figure 11 demonstrate six randomly selected videos from *LiveFood*. Each video, depending on the specific contents, may not necessarily contain all four predefined domains. Most importantly, we here stress that **we store no videos locally**, and we only provide the video links appearing in *LiveFood*, which means that the videos become unavailable if their creators delete them.

A.3 ANNOTATION QUALITY CONTROL

For subjective verification, each annotated video is reviewed by another experienced worker to identify whether each annotated segment satisfies the requirements as mentioned earlier in Table 1. An annotated video passes this check if both the annotator and reviewer reach an agreement on over 90% video segments. For example, the annotator marks ten clips out as highlights, and this video can pass this subjective check if another reviewer approves at least nine of them. During the objective check step, the reviewer is required to identify the start and end timestamps of the consensual clips mentioned above, and only clips with over 90% IoU (Intersection-over-Union) between the annotation of annotator and reviewer meet the demand of this check. The annotated videos are double-checked in a batch-wise manner, saying that 30 of every 100 annotated videos are selected randomly to be inspected subjectively and objectively. The threshold of qualification rate is set to 90%, which means that a batch of annotated videos is qualified when at least 27 of the 30 randomly selected videos pass the double-check mechanism. All annotators are trained with over one hundred

Figure 6: Samples of clips and domain annotations from `video_1`. The row from top to bottom illustrates vanilla clips, *presentation*, and *eating*, respectively. The faces appearing in video above are blurred to protect privacy.

Figure 7: Samples of clips and domain annotations from `video_2`. The row from top to bottom illustrates examples of *ingredients*, *cooking*, and *presentation*, respectively.

Figure 8: Samples of clips and domain annotations from `video_3`. The row from top to bottom illustrates examples of vanilla clips, *ingredients*, and *presentation*, respectively.

Figure 9: Samples of clips and domain annotations from video_4. The row from top to bottom illustrates examples of vanilla clips, *presentation*, and *eating*, respectively. The faces appearing in video above are blurred to protect privacy.

Figure 10: Samples of clips and domain annotations from video_5. The row from top to bottom illustrates examples of vanilla clips, *presentation*, and *eating*, respectively. The faces appearing in video above are blurred to protect privacy.

Figure 11: Samples of clips and domain annotations from video_6. The row from top to bottom illustrates examples of vanilla clips, *cooking*, and *eating*, respectively. The faces appearing in video above are blurred to protect privacy.

testing videos in advance of starting work officially, guaranteeing the quality of annotations. The whole collecting and checking process is costly in terms of time and money.

A.4 IN-HOUSE ANNOTATION TOOL

We use a private tool, *LiveLabel*, to annotate the selected videos. *LiveLabel* can be used for many complicated annotation tasks, such as video highlights annotation, image-text labeling, moment localization, and text-based video generation. There is no doubt that *LiveLabel* is a powerful tool for helping annotators to complete data preparation tasks in computer vision. However, we argue that due to the reasons of the tool's ownership, we can not make it publicly available. In the remainder of this section, we will detail the basic functions of *LiveLabel* in the usage of video highlights annotation, revealing its effects in easing labor-intensive data preparation tasks.

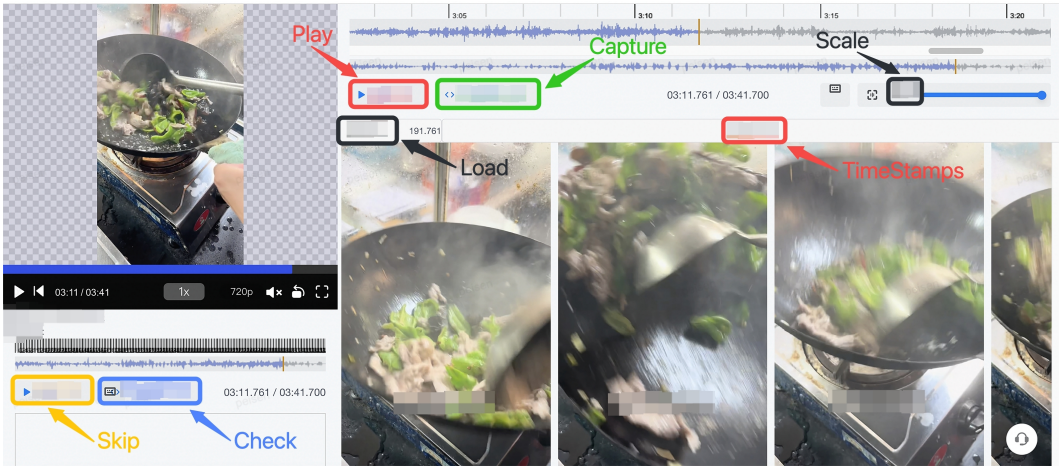


Figure 12: The annotation tools used in *LiveFood*.

As shown in the interface above, *LiveLabel* can pull videos from a pre-defined video pipeline automatically and parse these videos in a specific frame rate. On the left panel of Figure 12, the raw video is played at the user-specified speed. One can stop the video file if he/she encounters the candidates of video highlights. Following that, pushing the button of *capture* will demonstrate the detailed video contents in the frame-wise fashion, as shown in the right panel of the above interface. These frame-wise pictures help the annotators to check the fine timestamps of video highlights, mitigating the negative effects of incorrect annotations. After that, one can click the *check* button to select pre-defined domains or categories. In our experiments, we define the video clips into four domains, which are *ingredients*, *cooking*, *presentation*, and *eating*. These pieces of information can be input to the *LiveLabel* effortlessly, and during the usage, one just needs to select the target domain or category from the drop-down list. If there exist improper contents in videos, the annotator is asked to skip them using the *skip* button, and the *LiveLabel* will filter out these videos automatically, *i.e.*, remove them from *LiveFood*, to guarantee the quality of annotations and avoid privacy risks.

A.5 MORE COMPARISON RESULTS ON R-MNIST

In Table 4, we detail the comparison between our proposed GPE and existing incremental learning approaches on the R-MNIST benchmark. All methods employ the identical architecture, a perception model with two fully connected layers, to provide a fair comparison. In this section, we further investigate the convolution-based models with the training manner of GPE, aligning with the currently published schemes. Following Co^{2}L (Cha et al., 2021), we perform experiments on R-MNIST, which consists of 20 tasks. Since **no explicit** architectures of the convolution-based model are detailed in Cha et al. (2021), we use ResNet-18 (He et al., 2016) as the backbone by default. For a comprehensive comparison, we adopt the following three settings: 1) using no memory buffer, 2) employing a memory buffer with 200 samples, 3) employing a memory buffer with 500 samples. We borrow a part of the results from Co^{2}L (Cha et al., 2021). We present the references of methods shown in Table 7, which are ER (Rolnick et al., 2019), GEM (Lopez-Paz & Ranzato, 2017), A-GEM

(Chaudhry et al., 2019), FDR (Benjamin et al., 2019), GSS (Aljundi et al., 2019), HAL (Chaudhry et al., 2021), DER (Yan et al., 2021), DER++ (Yan et al., 2021), and Co^{2}L (Cha et al., 2021).

Table 7: Average classification accuracy on R-MNIST using convolution-based models.

Buffer	ER	GEM	A-GEM	FDR	GSS	HAL	DER	DER++	Co^{2}L	GPE (M_{d})
200	93.53	89.86	89.03	93.71	87.10	89.40	96.43	95.98	**97.90**	97.06
500	94.89	92.55	89.04	95.48	89.38	92.35	97.57	97.54	**98.65**	98.33

Noting that without using the memory buffer, our proposed GPE with ResNet-18 (He et al., 2016) backbone achieves 94.77% top-1 classification accuracy on the R-MNIST dataset. With resorting to the memory buffer, GPE yields better performance than its vanilla counterpart, generating performance improvements of 2.29% and 3.56%, respectively. Besides, we emphasize that in the experiments above, GPE employs a **random** memory buffer without specially designed schemes to replay the data samples. Moreover, we argue that GPE is proposed to serve as a strong baseline in practical applications instead of pursuing the *state-of-the-art* performance, thus we focus more on its scalability and expandability. Compared to the existing advanced incremental learning techniques such as DER (Yan et al., 2021) and Co^{2}L (Cha et al., 2021), GPE in the dynamic manner reports comparable results in a much more straightforward way, which is in good training efficiency and transfer ability. Unlike our proposed GPE, Co^{2}L (Cha et al., 2021) defends against forgetting using the memory buffer, strong augmentations, and distillation penalty. Although this scheme achieves *state-of-the-art* performance on several simple datasets, its applicability can not be guaranteed in complicated practical applications (*e.g.*, incremental VHD) since its specially designed augmentation schemes and hyper-parameters appearing in distillation procedure are not significantly stable when facing data from different domains. This can greatly influence the model's transfer ability and limit its usage in practical scenarios. In contrast, GPE well balances the model's performance and scalability, which is more suitable to serve as a starting point in many practical tasks.

A.6 Visualization of Learned Prototypes

Recall that we investigate the domain incremental learning task in this paper, and we present the *LiveFood* dataset, which consists of four domains, namely *ingredients*, *cooking*, *presentation*, and *eating*. The detailed visual annotation results are depicted in Appendix A.2. Although we emphasize the necessities and practicability of *LiveFood* in incremental VHD tasks, it can also be used as a challenging benchmark in both the vanilla video highlights detection and moment localization. Except for the newly presented dataset, we further propose GPE, a prototype-based framework, to learn incrementally with these domains, capturing new information from the fed data while remembering the observed knowledge. Taking the dynamic manner as samples, GPE inherits all learned highlight prototypes and limits their changing magnitude in the following tasks. Meanwhile, it develops a group of new prototypes to capture the novel information appearing in the newly fed data, *i.e.*, new domains. To provide a better understanding and analysis in-depth, we use t-SNE (Van der Maaten & Hinton, 2008) to demonstrate the incremental training process on *LiveFood*.

Concretely, we use GPE in the dynamic manner by default. We perform two groups of experiments, and the number of vanilla and highlight prototypes in each group is set to 40 and 80 respectively. We use scatters in different colors to indicate the various training tasks. $\mathcal{T}_1$, $\mathcal{T}_2$, $\mathcal{T}_3$, and $\mathcal{T}_4$ are used to indicate the prototypes learned at different stages. Figure 13 depicts the learning procedure.

In Figure 13, we set the number of learnable prototypes to 40 in the first row and 80 at the bottom. Taking the first (top left) two images as examples, GPE is trained to classify the vanilla and *presentation* clips. The scatters in gray are the negative prototypes, while that in pink are prototypes of *presentation*. During the second task, GPE aims to learn the characters of *eating* while still remembering that of *presentation*. Therefore, GPE limits the change of prototypes observed in $\mathcal{T}_1$ and develops a new group of prototypes in yellow, namely $\mathcal{T}_2$ positive in Figure 13. Here, we emphasize that prototypes in yellow (*i.e.*, $\mathcal{T}_2$ positive) are updated using the gradient descent method without constraints while that in pink (*i.e.*, $\mathcal{T}_1$ positive) are forbidden to change too much (cf. Eq.5). The following tasks are optimized similarly. Based on the depicted images, we can find that GPE adaptively adds some new positive prototypes with incremental tasks while modifying the position of negative prototypes which are set in a fixed quantity. Since we built the task as a binary classification problem, the goal of GPE is to reduce the discrepancy across different domains while boosting

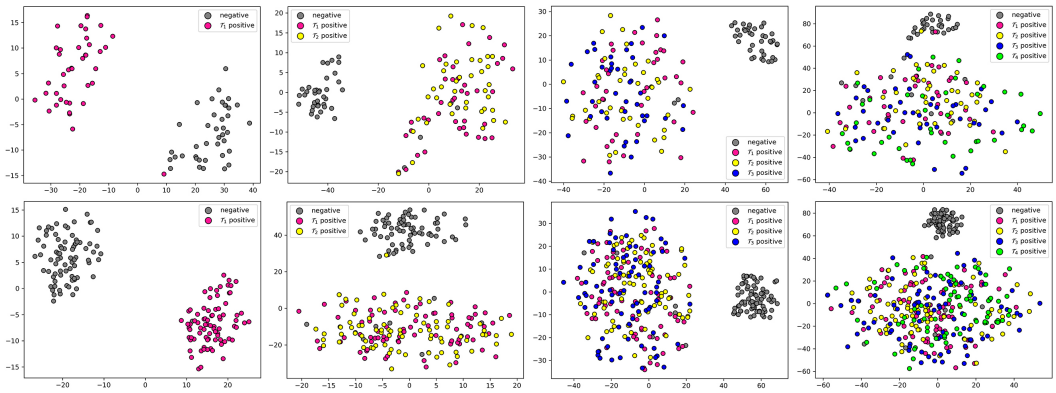


Figure 13: The learned prototypes of GPE across different training stages.

the distinction between the vanilla and highlight prototypes. The visualization results shown in Figure 13 are consistent with our expectation, depicting less discrepancy intra-domain while greater of that between the negative and positive prototypes. From another aspect, we allow the learned prototypes to change with a small magnitude for adapting to the new data, which means the inherited and the newly developed prototypes will be updated simultaneously, thus little discrepancy can be yielded among them.

A.7 *LiveFood* Release Agreement

One that requests permission to use the *LiveFood* dataset has to sign the following agreements and send a copy to us by email. After approval, we will provide you with the download links of our collected videos and annotations privately.

LiveFood Release Agreement

- The released *LiveFood* can be only used for non-commercial algorithm research and education. Researchers are forbidden to use this dataset for profitability or any infringement activities, including but not limited to advertising, selling, face-based applications, *etc*.
- The researchers applying for *LiveFood* admit that **no analysis** with respect to the **faces** appearing in this dataset is performed for protecting the privacy.
- The researchers take full responsibility for their usage of *LiveFood*. We keep the right to ask the users stop their illegal behavior and eliminate the negative influence.
- The researchers agree that we reserve the right to stop their usage and delete our provided data links, including their copy, if we find *LiveFood* is used for unexpected purposes.
- We have the right to ask the researchers applying for *LiveFood* to state their purposes and the potential effects clearly before getting access to the dataset.
- We hold **no** videos in *LiveFood* **locally**, and we only provide the **links** of videos appearing in *LiveFood*. These videos may become unavailable if their creators delete them.

Name: ____ **Email**: ____ **Organization**: ____ **Date**: ____ **Signature**: ____

Please state the purposes and potential effects of using this dataset in detail, and send it to us by email with a copy of the above agreements. Our email address will be made publicly available.