Exploring Domain Incremental Video Highlights Detection with the *LiveFood* Benchmark

Sen Pei[1], **Shixiong Xu**[2], **and Xiaojie Jin**[1]*

[1] ByteDance Inc.
[2] Institute of Automation, Chinese Academy of Sciences
{peisen, jinxiaojie}@bytedance.com, xushixiong2020@ia.ac.cn

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: *ingredients*, *cooking*, *presentation*, and *eating*. 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. Concerning the classic datasets, GPE also yields comparable performance as previous arts. The code is available at: https://github.com/ForeverPs/IncrementalVHD_GPE.

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, Grauman, and Sha 2018; Fajtl et al. 2018; Zhu et al. 2021), video highlights detection (VHD) (Yang et al. 2015; Xiong et al. 2019; Lei, Berg, and Bansal 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.

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

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, Song, and Cao 2016), PHD (Garcia del Molino and Gygli 2018), and QVHighlights (Lei, Berg, and Bansal 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 incremental learn-

*Corresponding author.

ing. 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, *e.g.*, *ingredients*, *cooking*, *presentation*, and *eating*. These related but distinctive domains provide a new test bed for incremental video highlights detection (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 its effectiveness.

The main contributions of this paper are summarized in three aspects 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 this research. *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 an end-to-end model for solving incremental VHD, *i.e.*, **G**lobal **P**rototype **E**ncoding (GPE). GPE incrementally identifies highlight/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 in practical scenarios.

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, Mei, and Rui (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, Berg, and Bansal 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. 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 by 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 and Cosgun 2018; Rolnick et al. 2019; Yan, Xie, and He 2021; Lange and Tuytelaars 2021), adopting regularization terms to constrain the change of model's weights or outputted logits (Kirkpatrick et al. 2016; Zenke, Poole, and Ganguli 2017; Schwarz et al. 2018), and performing parameter isolation to dedicate different model parameters for each task (Fernando et al. 2017; Mallya and Lazebnik 2018; Rosenfeld and 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, *i.e.*, different domains, 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.

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 $i \neq j$.

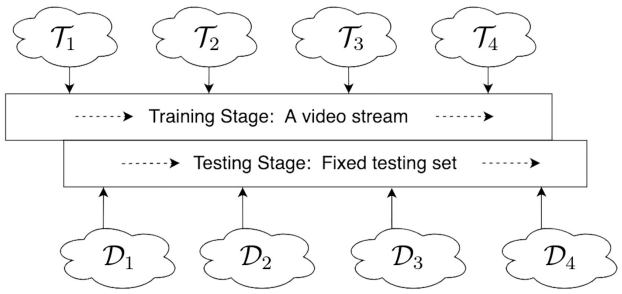


Figure 2: Conducting domain incremental video highlights detection on *LiveFood* dataset.

Specifically, in our collected $LiveFood$ dataset, we split the training videos into four disjoint sub-set, depicted as $\{\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3, \mathcal{T}_4\}$, 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 $\mathcal{C}_1 = \mathcal{S}_1$ 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.

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 4 (a) shows the duration distribution of the videos across all domains.

Highlights Annotation. We define four highlight domains that are generally presented in collected videos, namely *ingredients*, *cooking*, *presentation*, and *eating*. For each domain, a video clip is accepted as a satisfactory highlight if it meets the criteria in Figure 3. 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) 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.

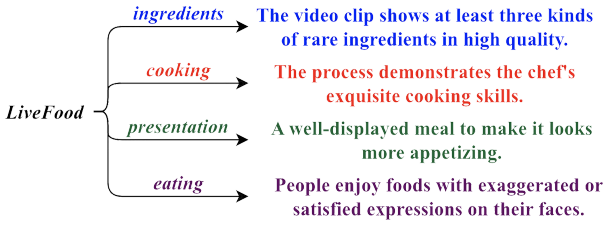
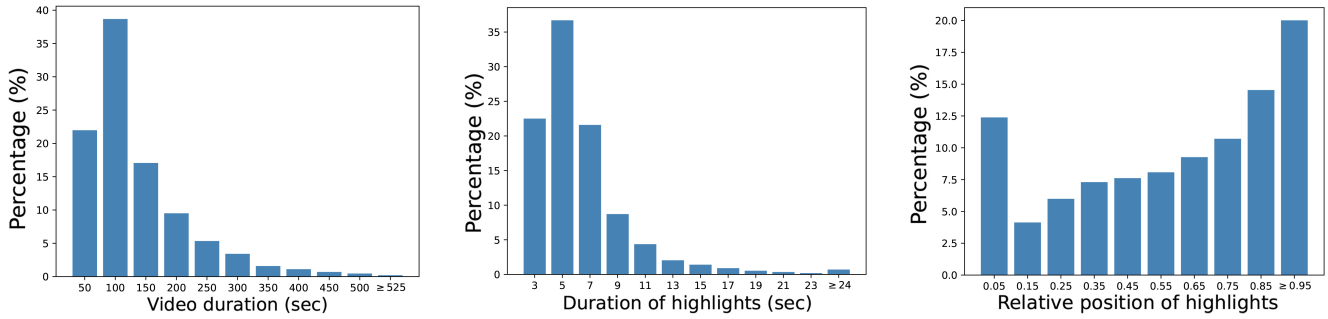


Figure 3: Basic descriptions of *LiveFood* domains.

Data Statistics. Figure 4 depicts the statistical results of our proposed *LiveFood*, including the distribution of highlights duration and the relative position of center timestamp with respect to each video. Besides, in Table 1, we also compare the proposed *LiveFood* with existing VHD datasets, such as SumMe (Gygli et al. 2014), YouTubeHighlights (Sun, Farhadi, and Seitz 2014), Video2GIF (Gygli, Song, and Cao 2016), PHD (Garcia del Molino and Gygli 2018), ActivityThumb (Yuan, Ma, and Zhu 2019), and QVHighlights (Lei, Berg, and Bansal 2021) for better illustrating the differences among them. As demonstrated in Table 1, 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



(a) (b) (c)

Figure 4: Statistical results of *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 with respect to corresponding videos. The highlights distribute evenly across whole videos, evidencing the good diversity of *LiveFood*.

Dataset	Year	Contents	Label domain/class	Human Annotated	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
ActivityThumb	2019	Activity	✗	✗	4K / 10K	60.7 / 8.7
QVHighlights	2021	Vlog / News	✗	✓	10.2K / 10.3K	150 / 24.6
LiveFood (ours)	2023	Gourmet	✓	✓	5.1K / 14.3K	136.5 / 6.4

Table 1: Comparison between the proposed *LiveFood* and existing datasets.

from the above datasets, *LiveFood* provides gourmet videos with finely annotated domain labels, making it suitable for domain-incremental VHD tasks.

Method

GPE aims to tackle forgetting while still improving by learning new concepts. As analyzed in Section , 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 (*i.e.*, M_{f} in Figure 5) or accommodation to new domain concepts (*i.e.*, M_{d} in Figure 5).

Architecture. Inspired by Carion et al. (2020), GPE employs the combination of convolution-based 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:

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

$$d_V = \min_{i=1:k} d(g_\phi(h), V_i), \tag{2}$$

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.

$$P_H = \frac{\exp(-d_H)}{\exp(-d_H) + \exp(-d_V)}. \tag{3}$$

Then, we use cross-entropy loss to optimize the model

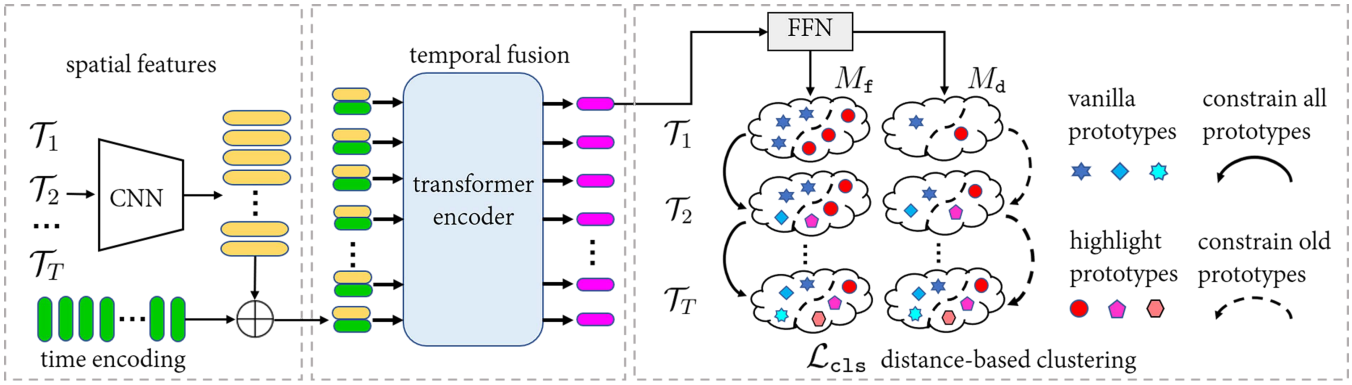


Figure 5: The proposed GPE framework. $\{\mathcal{T}_i\}_{i=1}^{T}$ indicates a training stream with T tasks. In the right-most part, M_f and M_d represents the fixed and dynamic modes of GPE. M_f defines the number of prototypes in advance and refines them across different stages. A restriction on the magnitude of change amplitude is imposed during learning (cf. Eq 6). M_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 2).

through gradient back-propagation:

$$\mathcal{L}_{cls} = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(P_H) + (1-y_i)\cdot \log(1-P_H) \quad (4)$$

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_f (cf. Figure 5) in this section, and the dynamic mode M_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 4 is abbreviated as $\mathcal{L}_{cls}(\theta, \phi, \pi)$. We expand the definition of distance measurement that evaluates 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:

$$d(\pi^{(t)}, \pi^{(t+1)}) = \frac{1}{k}\sum_{i=1}^{k}\sqrt{\sum_{j=1}^{m}(\pi_{i,j}^{(t)} - \pi_{i,j}^{(t+1)})^2} \quad (5)$$

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:

$$\begin{aligned} \min_{\theta,\phi,\pi} \quad & \mathcal{L}_{cls}(\theta, \phi, \pi) \\ \texttt{s.t.} \quad & d(\pi^{(T-1)}, \pi) \leq \gamma \end{aligned} \quad (6)$$

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 6 is transformed into the following manner:

$$\begin{aligned} S^{(T)} &= (\theta^{(T)}, \phi^{(T)}, \pi^{(T)}) \\ &= \max_{\lambda}\min_{\theta,\phi,\pi} \; L(\theta, \phi, \pi, \lambda) \\ &= \max_{\lambda}\min_{\theta,\phi,\pi} \; \mathcal{L}_{cls}(\theta, \phi, \pi) + \lambda[d(\pi^{(T-1)}, \pi) - \gamma] \end{aligned} \quad (7)$$

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 as following:

$$\begin{aligned} \theta &\leftarrow \theta - \eta\frac{\partial \mathcal{L}_{cls}(\theta, \phi, \pi)}{\partial \theta} \\ \phi &\leftarrow \phi - \eta\frac{\partial \mathcal{L}_{cls}(\theta, \phi, \pi)}{\partial \phi} \\ \pi &\leftarrow \pi - \eta\frac{\partial L(\theta, \phi, \pi, \lambda)}{\partial \pi} \\ \lambda &\leftarrow \max\{\lambda + \eta[d(\pi^{(T-1)}, \pi) - \gamma], 0\} \end{aligned} \quad (8)$$

where η is the learning rate of trainable parameters and λ is the multiplier in the dual step. The incremental training and inference pipeline is summarized in Appendix 1.

Experiment

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

Experimental Setup

Data and Evaluation Protocol. *LiveFood* contains 4928 videos for training and 261 videos for testing. We randomly

mAP	Lb	SI	oEWC	ER*	DER*	Ub	GPE (M_{f})	GPE (M_{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	33.43	33.90	**34.05**

Table 2: Comparison of GPE with existing incremental learning methods on *LiveFood*. We evaluate their frame-wise mAP performance. The superscript of * (star) indicates that the method uses memory buffer to boost its memory ability.

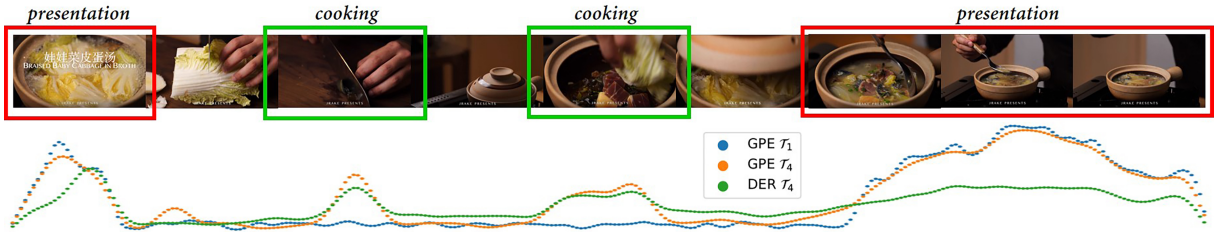


Figure 6: The observed highlights across different training stages. The curves in different color indicate the highlight scores, which are predicted by the corresponding models within several training stages.

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, Mei, and Rui 2016; Xiong et al. 2019).

Comparable baselines. We employ both regularization-based methods, such as SI (Zenke, Poole, and Ganguli 2017) and oEWC (Schwarz et al. 2018), and replay methods, such as ER (Rolnick et al. 2019) and DER (Yan, Xie, and He 2021), as our comparable baselines. Besides, we also set the lower bound (Lb) and upper bound (Ub) of incremental learning. Please refer to Appendix for more details. Other techniques are introduced during the context.

Main Result

Comparison with existing IL methods. The experimental results are depicted in Table 2. We highlight the upper bound results with gray background. M_{f} and M_{d} represent the fixed and dynamic mode of GPE. The superscript of * (star) indicates that the method uses memory buffer to boost its memory ability. It can be observed from Table 2 that the vanilla GPE (M_{f}) surpasses the lower bound with an improvement of 2.91% mAP. Compared to the classic IL methods SI (Zenke, Poole, and Ganguli 2017) and oEWC (Schwarz et al. 2018), 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, Xie, and He (2021), GPE achieves 1.06% and 2.71% mAP gain compared to DER (Yan, Xie, and He 2021) and ER (Rolnick et al. 2019). The above results clearly demonstrate the effectiveness of GPE in tackling the incremental VHD task.

Visualization of highlight scores across training stages. We investigate the effects of observed prototypes across different training stages. In Figure 6, 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 (Yan, Xie, and He 2021). In the curves shown in Figure 6, the blue and orange points indicate the highlight scores of each frame predicted by GPE, *i.e.*, P_H in Eq 3, 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_{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). Results depicted in Table 3 demonstrate that the dynamic GPE surpasses most conventional methods, including the regularization-based and replay methods. Notably, the dynamic GPE achieves 85.42%

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	<u>85.22</u>	79.50	84.02	90.04	$85.42_{\pm 0.13}$	$\mathbf{90.17}_{\pm 0.25}$

Table 3: Average classification accuracy on R-MNIST using linear classifiers.

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**	<u>97.06</u>
500	94.89	92.55	89.04	95.48	89.38	92.35	97.57	97.54	**98.65**	<u>98.33</u>

Table 4: Average classification accuracy on R-MNIST using convolution-based models.

average accuracy across 20 tasks, comparable to ER (Rolnick et al. 2019) and FDR (Benjamin, Rolnick, and Körding 2019) while outperforming other methods with a considerable margin. We further adopt the identical replay schemes as DER (Yan, Xie, and He 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. Besides, we further investigate the extensibility of GPE to stronger backbones, *e.g.*, the convolution network as in Cha, Lee, and Shin (2021). We use ResNet-18 (He et al. 2016) as the backbone for GPE and all other competing methods. The result is depicted in Table 4 and the detailed explanation of this part is attached in Appendix 18. To summary, when **no** memory buffer is applied, GPE with ResNet-18 (He et al. 2016) backbone achieves **94.77%** (cf. Appendix Table 4) top-1 classification accuracy on the R-MNIST dataset. By using a memory buffer of 200 samples and 500 samples, GPE boosts its performance by 2.29% and 3.56%, respectively. These results achieve the comparable *state-of-the-art* performance.

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 under-fitting. 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. If the dimension of features outputted by the feedforward layer (FFN) in transformer encoder is not significantly high, the extra training cost yielded by prototypes is not unbearable since the flops has linear complexity with the feature dimension.

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 2. 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, resulting in a near 1.0% performance drop. Consequently, we set k and γ to 40 and 5 by default.

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 5: Ablations on the initial number of prototypes k.

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

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

Conclusion

In this paper, we introduce a new task: incremental video highlights 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, *i.e.*, *ingredients*, *cooking*, *presentation*, and *eating*. The data collection procedure has been reviewed and approved by an institutional review board (IRB) equivalent committee. We also propose a novel model named **G**lobal **P**rototype **E**ncoding (GPE) to learn incrementally to adapt to new highlight domains. Extensive experiments 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

Aljundi, R.; Lin, M.; Goujaud, B.; and Bengio, Y. 2019. Gradient based sample selection for online continual learning. In *NeurIPS*.

Badamdorj, T.; Rochan, M.; Wang, Y.; and Cheng, L. 2021. Joint Visual and Audio Learning for Video Highlight Detection. In *ICCV*.

Badamdorj, T.; Rochan, M.; Wang, Y.; and Cheng, L. 2022. Contrastive Learning for Unsupervised Video Highlight Detection. In *CVPR*.

Benjamin, A. S.; Rolnick, D.; and Körding, K. P. 2019. Measuring and regularizing networks in function space. In *ICLR*.

Bhattacharya, U.; Wu, G.; Petrangeli, S.; Swaminathan, V.; and Manocha, D. 2021. HighlightMe: Detecting Highlights From Human-Centric Videos. In *ICCV*.

Buzzega, P.; Boschini, M.; Porrello, A.; Abati, D.; and Calderara, S. 2020. Dark Experience for General Continual Learning: a Strong, Simple Baseline. In *NeurIPS*.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-End Object Detection with Transformers. In *ECCV*.

Cha, H.; Lee, J.; and Shin, J. 2021. Co^{2}L: Contrastive Continual Learning. In *ICCV*.

Chaudhry, A.; Gordo, A.; Dokania, P. K.; Torr, P. H. S.; and Lopez-Paz, D. 2021. Using Hindsight to Anchor Past Knowledge in Continual Learning. In *AAAI*.

Chaudhry, A.; Ranzato, M.; Rohrbach, M.; and Elhoseiny, M. 2019. Efficient Lifelong Learning with A-GEM. In *ICLR*.

Fajtl, J.; Sokeh, H. S.; Argyriou, V.; Monekosso, D.; and Remagnino, P. 2018. Summarizing Videos with Attention. In *ACCV*.

Fernando, C.; Banarse, D.; Blundell, C.; Zwols, Y.; Ha, D.; Rusu, A. A.; Pritzel, A.; and Wierstra, D. 2017. PathNet: Evolution Channels Gradient Descent in Super Neural Networks. In *CoRR*.

Garcia del Molino, A.; and Gygli, M. 2018. PHD-GIFs: Personalized Highlight Detection for Automatic GIF Creation. In *ACM MM*.

Gygli, M.; Grabner, H.; Riemenschneider, H.; and Van Gool, L. 2014. Creating Summaries from User Videos. In *ECCV*.

Gygli, M.; Song, Y.; and Cao, L. 2016. Video2GIF: Automatic Generation of Animated GIFs from Video. In *CVPR*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*.

Isele, D.; and Cosgun, A. 2018. Selective Experience Replay for Lifelong Learning. In *AAAI*.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N. C.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; Hassabis, D.; Clopath, C.; Kumaran, D.; and Hadsell, R. 2016. Overcoming catastrophic forgetting in neural networks. In *CoRR*.

Lange, M. D.; Aljundi, R.; Masana, M.; Parisot, S.; Jia, X.; Leonardis, A.; Slabaugh, G. G.; and Tuytelaars, T. 2022. A Continual Learning Survey: Defying Forgetting in Classification Tasks. In *TPAMI*.

Lange, M. D.; and Tuytelaars, T. 2021. Continual Prototype Evolution: Learning Online from Non-Stationary Data Streams. In *ICCV*.

Lei, J.; Berg, T. L.; and Bansal, M. 2021. Detecting Moments and Highlights in Videos via Natural Language Queries.

Liu, M.; Wang, X.; Nie, L.; Tian, Q.; Chen, B.; and Chua, T. 2018. Cross-modal Moment Localization in Videos. In *ACM MM*.

Liu, Z.; Mao, H.; Wu, C.-Y.; Feichtenhofer, C.; Darrell, T.; and Xie, S. 2022. A ConvNet for the 2020s.

Lopez-Paz, D.; and Ranzato, M. 2017. Gradient Episodic Memory for Continual Learning. In *NeurIPS*.

Mallya, A.; and Lazebnik, S. 2018. PackNet: Adding Multiple Tasks to a Single Network by Iterative Pruning. In *CVPR*.

Potapov, D.; Douze, M.; Harchaoui, Z.; and Schmid, C. 2014. Category-Specific Video Summarization. In *ECCV*.

Rebuffi, S.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. iCaRL: Incremental Classifier and Representation Learning. In *CVPR*.

Rodriguez, C.; Marrese-Taylor, E.; Saleh, F. S.; LI, H.; and Gould, S. 2020. Proposal-free Temporal Moment Localization of a Natural-Language Query in Video using Guided Attention. In *WACV*.

Rolnick, D.; Ahuja, A.; Schwarz, J.; Lillicrap, T. P.; and Wayne, G. 2019. Experience Replay for Continual Learning. In *NeurIPS*.

Rosenfeld, A.; and Tsotsos, J. K. 2020. Incremental Learning Through Deep Adaptation. In *TPAMI*.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. In *IJCV*.

Saquil, Y.; Chen, D.; He, Y.; Li, C.; and Yang, Y.-L. 2021. Multiple Pairwise Ranking Networks for Personalized Video Summarization. In *ICCV*.

Schwarz, J.; Czarnecki, W. M.; Luketina, J.; Grabska-Barwinska, A.; Teh, Y. W.; Pascanu, R.; and Hadsell, R. 2018. Progress & Compress: A scalable framework for continual learning. In *CoRR*.

Song, Y.; Vallmitjana, J.; Stent, A.; and Jaimes, A. 2015. TVSum: Summarizing web videos using titles. In *CVPR*.

Sun, M.; Farhadi, A.; and Seitz, S. M. 2014. Ranking Domain-Specific Highlights by Analyzing Edited Videos. In *ECCV*.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. In *JMLR*.

Xiong, B.; Kalantidis, Y.; Ghadiyaram, D.; and Grauman, K. 2019. Less Is More: Learning Highlight Detection From Video Duration. In *CVPR*.

Yan, S.; Xie, J.; and He, X. 2021. DER: Dynamically Expandable Representation for Class Incremental Learning. In *CVPR*.

Yang, H.; Wang, B.; Lin, S.; Wipf, D. P.; Guo, M.; and Guo, B. 2015. Unsupervised Extraction of Video Highlights via Robust Recurrent Auto-Encoders. In *ICCV*.

Yao, T.; Mei, T.; and Rui, Y. 2016. Highlight Detection with Pairwise Deep Ranking for First-Person Video Summarization. In *CVPR*.

Yuan, Y.; Ma, L.; and Zhu, W. 2019. Sentence Specified Dynamic Video Thumbnail Generation. In *ACM MM*.

Zenke, F.; Poole, B.; and Ganguli, S. 2017. Continual Learning Through Synaptic Intelligence. In *ICML*.

Zhang, K.; Grauman, K.; and Sha, F. 2018. Retrospective Encoders for Video Summarization. In *ECCV*.

Zhang, S.; Peng, H.; Fu, J.; and Luo, J. 2020. Learning 2d temporal adjacent networks for moment localization with natural language. In *AAAI*.

Zhu, W.; Lu, J.; Li, J.; and Zhou, J. 2021. DSNet: A Flexible Detect-to-Summarize Network for Video Summarization. In *TIP*.