Long-Tail Temporal Action Segmentation with Group-wise Temporal Logit Adjustment

Zhanzhong Pang[1], Fadime Sener[2], Shrinivas Ramasubramanian[3], and Angela Yao[1]

[1] National University of Singapore
[2] Meta Reality Labs
[3] Carnegie Mellon University
{pang, ayao}@comp.nus.edu.sg, famesener@meta.com, shrinivr@cs.cmu.edu

Abstract. Procedural activity videos often exhibit a long-tailed action distribution due to varying action frequencies and durations. However, state-of-the-art temporal action segmentation methods overlook the long tail and fail to recognize tail actions. Existing long-tail methods make class-independent assumptions and struggle to identify tail classes when applied to temporal segmentation frameworks. This work proposes a novel group-wise temporal logit adjustment (G-TLA) framework that combines a group-wise softmax formulation while leveraging activity information and action ordering for logit adjustment. The proposed framework significantly improves in segmenting tail actions without any performance loss on head actions. Source code is available[4].

Keywords: Temporal action segmentation · Procedural video understanding · Long-tail recognition · Logit adjustment

1 Introduction

Temporal action segmentation [25,30,43] partitions procedural activity sequences into multiple segments, each corresponding to a specific action class as depicted in Fig. 1. There are two sources of tail in temporal action segmentation datasets. First, procedural videos often exhibit a long-tail distribution of action segments [12], with infrequent actions forming the tail. For instance, in Fig. 1, when *"making tea"*, *'add teabag'* is indispensable, while *'spoon sugar'* and *'stir tea'* are optional. The temporal action segmentation problem is mainly framed as a frame-wise classification task [14, 42, 50], which leads to a second source of long-tail due to the varying durations of actions in procedural videos. For example, *'pour water'* in Fig. 1 is longer and has more frames than *'spoon sugar'*. The segment- & frame-wise imbalances can be quite extreme, as depicted in Fig. 2.

The imbalance problem has been overlooked in temporal action segmentation literature [12,14,17,42,50], leading to poor performance on tail classes[5]. Our

[4] https://github.com/pangzhan27/GTLA
[5] ASFormer [50], DiffAct [33] have zero accuracy on 5 and 4 of 48 classes, respectively.

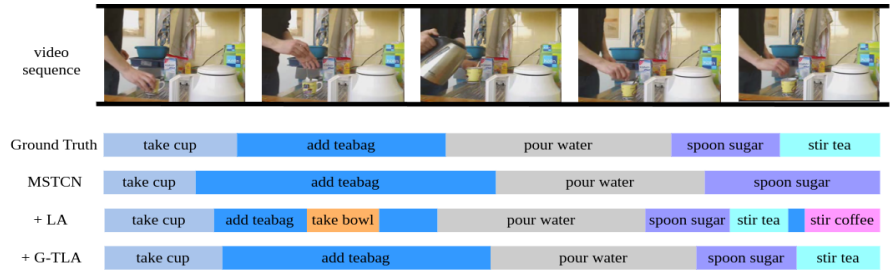
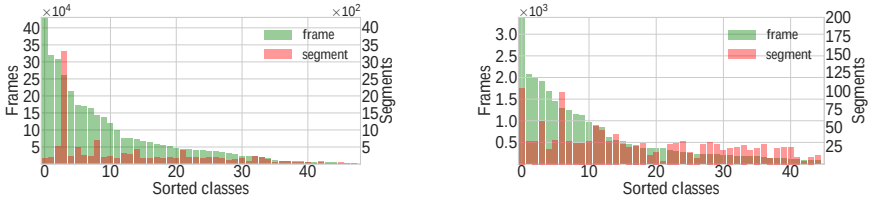


Fig. 1: *"Making tea"*, with temporal segments indicated by colored bars. The tail action *'stir tea'* is recognized by Logit adjustment (LA) and our G-TLA but not by the MSTCN backbone. However, LA overlooks the action order and activity, resulting in activity-irrelevant false positives such as *'take bowl'* & *'stir co ee'*, and temporally illogical false positives like *'add teabag'* occurring after *'stir tea'*.

paper fills a significant gap by addressing the long-tail problem in temporal segmentation for the first time. Long-tail learning has mainly been investigated in image [11,22,35] and video classification [37,52]. Temporal segmentation differs, however, because actions are temporally correlated. Yet conventional long-tail learning solutions such as re-sampling [18, 20], loss re-weighting [11, 32], and logit adjustment (LA) [35,46] make class independence assumptions. This compromises the learned temporal dependencies within the base models [30,50]; for example, LA incorrectly predicts *'add teabag'* after *'stir tea'* in Fig. 1. Moreover, for action segmentation, both frame- and segment-level performance is measured [14]. Balancing head and tail classes, as well as frame- and segment-wise performance are two challenging trade-offs. Conventional methods are ineffective in addressing the frame and segment trade-offs. For example, LA in Fig. 1 introduces irrelevant classes of *'stir coffee'* and *'take bowl'* from other activities, increasing false positives and over-segmentation.

In procedural activities, some actions like *'spoon sugar'* are shared across activities of *"making tea"* and *"making coffee"*; others are activity-specific, *e.g.* action *'stir tea'* occurs only in the activity of *"making tea"*. Additionally, actions follow certain ordering: *'stir tea'* always follows *'pour water'*, even if other actions like *'spoon sugar'* occur in between. Based on these observations, we propose a novel Group-wise Temporal Logit Adjustment (G-TLA) framework that encodes action order and activity information to address the long tail problem. Our framework consists of group-wise classification based on the activity label and temporal logit adjustment based on action ordering priors. Our method enhances tail recognition and mitigates over-segmentation by reducing false positives as shown in Fig. 8, including those from irrelevant activities and violating temporal dependencies. Our contributions can be summarized as:

- the first method to address long-tail temporal action segmentation, along with our novel group-wise temporal logit adjustment (G-TLA) approach.
- leveraging class interdependencies to reduce false positives from long-tail learning, reducing over-segmentation.



(a) Breakfast. **(b)** YouTube

Fig. 2: Temporal action segmentation datasets exhibit a long-tail distribution of actions due to varying frequencies of actions and action durations.

- proposing new evaluation metrics to better reflect tail class performance, otherwise obscured by standard metrics.
- extensive evaluations on five datasets [1, 16, 26, 39, 44], outperforming state-of-the-art backbones and standard long-tail learning approaches.

2 Related works

Temporal Action Segmentation. Early approaches [2, 9, 15] employed handcrafted features to model frame-wise dependencies while using HMMs [27] or RNNs [36,41] to capture long-term dependencies. Subsequent methods leveraged pre-computed video features, such as I3D [8] and employed TCNs [14, 28–30, 42], transformers [50] and diffusion models [33] to learn frame- and action-wise relationships. For a comprehensive overview of temporal action segmentation, we refer to a recent survey [12]. To our knowledge, we are the first to address the long-tail problem in video sequences for temporal action segmentation [12].
Long-Tail Learning. Long-tail methods can be broadly split into data and algorithm-level methods. Data-level methods either oversample the tail [4, 5, 40] or undersample the head [13, 19]. However, both approaches have drawbacks: oversampling can lead to tail overfitting, while undersampling is suboptimal for video datasets, as they often have limited samples. Additionally, applying existing data-level methods naively at the frame level is unsuitable for temporal action segmentation, as it requires preserving the sequence entirely as input.

Algorithm-level methods include cost-sensitive learning, post-hoc adjustment, and ensembling techniques. Cost-sensitive learning balances loss functions by reweighting different classes or samples [11,21,32,38,48], or by adjusting the logits to enlarge margins for tail classes [7,35,45,46,53]. Post-hoc adjustment improves predictions for rare classes after training, *e.g.* by normalizing the classification weights [22, 23, 51] or modifying the classification threshold [10, 24]. Ensembling combines multiple modules or experts, each specializing in different class distributions [47, 54] or subsets [6, 31]. A key assumption in these methods is class independence. However, in action segmentation, videos can be assumed to be *i.i.d.*, while the actions within each video exhibit dependencies. Our work is the first to consider these dependencies in designing a long-tailed solution.

3 Preliminaries

3.1 Temporal Action Segmentation

Temporal action segmentation maps frame-wise representations, $\mathcal{X}$, to action labels, $\mathcal{Y} = [L] = \{1, 2, \ldots, L\}$. Consider a video sample, $\{X, Y\}$: $X = \{x_t\} \in \mathbb{R}^{D \times T}, Y = \{y_t\} \in [L]^T$, where D is the feature dimension, T is the number of frames in a video, t is the frame index, L is the number of classes. In line with previous works [14, 42, 50], we use pre-computed video representations [8] as X.

Consider a classifier, denoted as f, which takes the entire sequence, X, as input to learn temporal correlations among frames and outputs a sequence of action labels, Y. Classifier f is a neural network such as MSTCN [14] or AsFormer [50]. Classifier $f : \mathcal{X} \to \mathcal{Y}$ is trained by minimizing a frame-wise cross-entropy loss:

$$\mathcal{L}_{cls} = \frac{1}{T} \sum_{t} - \log \hat{p}(y_t), \tag{1}$$

where $\hat{p}(y_t)$ is the estimated probability for the ground truth label y_t of frame x_t. A smoothing loss is commonly applied with threshold δ to encourage smooth transitions between frames. Following prior works [14], we set δ to 4.

$$\begin{aligned} \mathcal{L}_{sm} &= \frac{1}{TL} \sum_{t,c} \hat{\triangle}^2_{t,c}, \quad \hat{\triangle}_{t,c} = \begin{cases} \triangle_{t,c}, & \triangle_{t,c} \leq \delta \\ \delta \quad , & \text{otherwise} \end{cases} \\ \triangle_{t,c} &= |\log \hat{p}(c_t) - \log \hat{p}(c_{t-1})|, \ c \in [1, \cdots, L], \end{aligned} \tag{2}$$

where $\hat{p}(c_t)$ is the estimated probability of class c at time t.

3.2 Logit Adjustment

Given a classification problem with data-label pairs $\{x, y\}$, the standard training objective is to learn a model f that minimizes the expected error $\mathbf{E}_{x,y}\, \mathrm{Err}(f(x), y)$. When the label distribution is highly skewed, the learning process minimizes a skewed error, prioritizing classes with more samples. In such scenarios, a balanced error $\mathbf{E}_{x|y}\, \mathrm{Err}(f(x), y)$ that averages the per-class loss [3, 34] is more suitable. Given the prior $p(c)$ and the posterior $p(c\,|\,x)$ of class c, the optimal classifier for minimizing the balanced error [10] takes the form

$$\arg\max_c \frac{p(c \mid x)}{p(c)} \approx \arg\max_c s_c(x) - \log p(c) \tag{3}$$

where a neural network typically estimates $p(c\,|\,x)$ as a logit $s_c(x)$. For simplicity, we omit the logit's dependency on the neural network's parameters.

Eq. (3) is the vanilla logit adjustment [35], which incorporating the prior during inference. Logit adjustment can be further incorporated into training by enforcing a class prior offset while learning the logits [35].

$$\tilde{s}_c(x) = s_c(x) + \tau \log p(c), \quad \tilde{p}(c|x) = \mathrm{softmax}(\tilde{s}_c(x)) \tag{4}$$

where $s_c(x)$ is the output logit, $\tilde{s}_c(x)$ is the adjusted logit, $\tilde{p}(c|x)$ is the predicted probability of class c after adjusting logit and used in the cross-entropy loss(we use p to represent the prior, $\hat{p}$ for predicted probability, and $\tilde{p}$ for probability after adjusting logit). τ controls the trade-off between minimizing balanced and skewed errors. Introducing the adjustment adds a per-class margin into the softmax, shifting the decision boundary away from tail classes as in Eq. (5).

$$\mathcal{L}(y, f(x)) = -\log \frac{e^{s_y(x)+\tau \log p(y)}}{\sum_c e^{s_c(x)+\tau \log p(c)}} = \log \left[1 + \sum_{c \neq y} e^{s_c(x) - s_y(x) + \tau \frac{\log p(c)}{\log p(y)}} \right] \quad (5)$$

4 Methodology

Our method leverages action order and activity information to address the long-tail problem in action segmentation. We begin by introducing the action inter-dependencies in Sec. 4.1 and describe how to solve such dependencies through group-wise classification in Sec. 4.2 and temporal logic adjustment in Sec. 4.3. Details of training and inference are presented in Sec. 4.4 and Sec. 4.5.

4.1 Action Inter-Dependencies

The independent assumption of prior $p(c)$ in Eq. (4) does not hold for actions in videos. In procedural activities, actions within a sequence interact. Accurately estimating the prior of action c conditioned on sequence Y, $p(c|Y)$ is challenging due to varying action orders and limited training samples. Notably, the class distribution $p(c)$ is activity, a, dependent; naively using $p(c)$ leads to activity-irrelevant false positives as in Fig. 1. Therefore, we propose a relaxed solution for the temporal prior as $p(c|a)$ to condition on activity label a. As the video sequences have a loose ordering of actions, we assume that action classes are conditionally independent given the activity label. We further exploit useful action ordering in Sec. 4.3.

Given an input sequence, X, and its activity label, a, frame predictions are independent while frame representations interact [14,50], *i.e.* frame prediction $\hat{y}_t$ is based on the sequence representation X instead of a single frame representation x_t. Then, the logit adjustment for frame t, according to Eq. (4), is:

$$s_{c,t}(X) + \tau \log p(c \mid a), \quad (6)$$

where $s_{c,t}(X)$ is the logit of class c at time t.

4.2 Group-wise Classification

To leverage the conditional prior $p(c|a)$, we propose a hierarchical classifier by categorizing sequences into disjoint groups and assigning actions within each sequence to the corresponding group. This grouping strategy enables group-wise

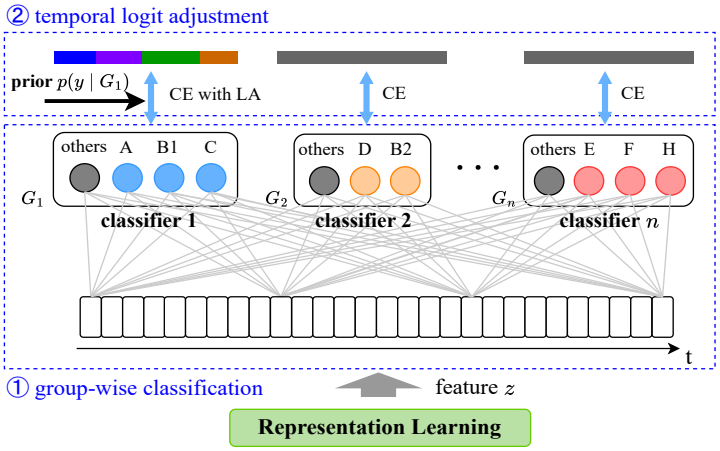


Fig. 3: Our group-wise temporal logit adjustment framework consists of group-wise classification and temporal logit adjustment within the respective group. The temporal logit adjustment is only applied to the target group(G_1 in this illustration).

action classification and logit adjustment based on the group's prior. Additionally, grouping ensures no incompatible actions in each activity group. For example, the action *'stir coffee'* does not occur in activities like *"making tea"*. The incompatibility causes zero conditional probability $p(c|a) = 0$, posing a numerical problem for logit adjustment. Adding a small value to $p(c|a)$ does not solve the numerical problem. Instead, it converts activity-incompatible classes to tail classes. These classes are subsequently overemphasized and result in more activity-irrelevant false positives.

We denote the group set as $\mathbf{G}$. Each group may contain activities with shared actions. While a straightforward approach is to group based on the activity label, we also explore sequence clustering without activity labels based on the KL-divergence of action frequency distributions(see Supplementary for details). For each group, we introduce an auxiliary class, *'others'*, corresponding to actions that do not belong to the current group. Classes shared across different groups are treated as separate classes. For example, *'spoon sugar'* occurs in both *"making coffee"* and *"making tea"* which are different groups G_1 and G_2, then *'spoon sugar'* is treated as different classes $y^{(1)}$ and $y^{(2)}$. Non-shared classes, like *'stir tea'*, are labeled as *'stir tea'* in G_2, and *'others'* in the remaining groups.

Our group-wise classification strategy is illustrated in Fig. 3. The final feature layer z_t at time t is fed into n classifiers, where n is the number of groups in $\mathbf{G}$. These classifiers generate predictions concurrently based on the last feature layer. For the $i^{\text{-th}}$ group, the predicted logit of class c at time t is computed as

$$s_{c,t}^{(i)}(X) = \sum_j z_t[j] \cdot W_{j,c}^{(i)} + b^{(i)} \tag{7}$$

where $W^{(i)}$ and $b^{(i)}$ are the classifier weights and bias for group i.

The overall loss is the sum of two cross-entropies: one for the actions in the target group and one for the auxiliary class for non-target groups:

$$\mathcal{L}_G = \alpha_k \frac{1}{T} \sum_t -\log \hat{p}(y_t^{(k)}) + \eta \sum_{i \neq k}^{n} \frac{1}{T} \sum_t -\log \hat{p}(o_t^{(i)}) \tag{8}$$

where k is the current sequence's group, $\hat{p}(y_t^{(k)})$ represents the predicted probability of the truth label y_t in the target group k, $\hat{p}(o_t^{(i)})$ is the predicted probability of class *'others'* in the non-targeted group. η is a hyperparameter that balances the losses between target and non-target groups and α_k represents a pre-computed reweighting factor designed to balance the bias arising from group size, *i.e.* the number of sequences in each group. From a gradient analysis perspective [46], a small η helps reduce the suppression of tail classes by down-weighting the gradients from negative samples of the class *others*. However, if η is too small, it may hinder group identification during inference.

Group-wise classification offers several advantages. Separating semantically similar actions into different groups mitigates confusion, *e.g.* distinguishing between *'take plate'* and *'take cup'* becomes easier, despite sharing a verb. Additionally, it reduces false positives from irrelevant classes, ensuring that only group-compatible actions are predicted, preventing implausible predictions.

4.3 Temporal Logit Adjustment

Our group-wise framework simplifies frame-wise classification. With group information we can substitutes $p(c|a)$ in Eq. (6) with $p(c|G_k)$. However, there is still valuable sequential ordering information. For example, the action *'stir tea'* always follows *'pour water'*, even if other actions *e.g.* *'spoon sugar'* occur in between. This ordering information can also be incorporated into the logit adjustment. Therefore, we adopt a fine-grained approach by tailoring the adjustment to each class c and frame t within the target group G_k, as follows:

$$s_{c,t}^{(k)}(X) + \tau \mathcal{T}_{c,t}^{(k)}(X) \log p(c \mid G_k), \tag{9}$$

where the temporal factor $\mathcal{T}_{c,t}^{(k)}(X)$ refines the logit adjustment based on the temporal bounds for class c in the target group G_k.

Temporal Factor. For an action c, we define $S_{bf}[c]$ as the set of actions that must precede c, and $S_{af}[c]$ that must follow c. These two sets are exclusive and can be derived from the training data. Given a sequence X from group G_k, we can utilize the two sets to determine the temporal bounds $[t_1(c,X), t_2(c,X)]$ within which action c can occur, where $t_1(c,X)$ and $t_2(c,X)$ are determined by the latest and earliest occurrence time of classes in $S_{bf}[C]$ and $S_{af}[C]$, respectively.

$$t_1(c,X) = \begin{cases} \max_t y_t \in S_{bf}[c], \text{ if } S_{bf}[c] \neq \emptyset \\ 0, \text{ otherwise} \end{cases}, t_2(c,X) = \begin{cases} \min_t y_t \in S_{af}[c], \text{ if } S_{af}[c] \neq \emptyset \\ T, \text{ otherwise,} \end{cases} \tag{10}$$

where y_t is the ground truth label of X at time t.

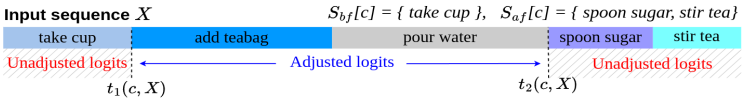


Fig. 4: Illustration of temporal logit adjustment for class $c =$ *'add teabag'*. The adjustment only occurs within the temporal bounds.

Temporal factor adjusts the logit for class c in the target group G_k based on these temporal bounds. Logits within the temporal bounds are adjusted normally, while those outside the bounds are left unadjusted to prevent false positives that violate the temporal prior as in Fig. 4. This can be formalized as:

$$\mathcal{T}_{c,t}^{(k)}(X) = \begin{cases} 1, & \text{if } t \in [t_1(c,X), t_2(c,X)] \\ \frac{\log p(y_t|G_k)}{\log p(c|G_k)}, & \text{otherwise,} \end{cases} \tag{11}$$

where *otherwise*, we use the ratio of the prior for label y_t and class c to ensure consistent adjustment with the target label, $p(y_t \mid G_k)$. This prevents misclassification due to adjustment between y_t and c, avoiding violating temporal prior.

4.4 Overall loss

With group-wise classification and temporal logit adjustment, we can reframe the classification loss Eq. (8) as

$$\mathcal{L}_{GTLA} = \alpha_k \frac{1}{T} \sum_t -\log \tilde{p}(y_t^{(k)}) + \eta \sum_{i \neq k}^{n} \frac{1}{T} \sum_t -\log \hat{p}(o_t^{(i)}), \tag{12}$$

where predicted probabilities $\tilde{p}$ for the target group k is based on adjusted logits in Eq. (9). For non-target groups i, logit adjustment is not performed; instead, the naive logit $s_{o,t}^{(i)}(X)$ of class *'others'* is used to calculate the probability.

The training loss $\mathcal{L}$ combines group-wise logit adjustment loss $\mathcal{L}_{GTLA}$ and smooth loss $\mathcal{L}_{sm}$, $\mathcal{L} = \mathcal{L}_{GTLA} + \lambda \mathcal{L}_{sm}$, with λ as the trade-off hyper-parameter.

4.5 Inference

The group label of a sequence is unknown during inference. We use the prediction for *'others'* to identify the group. Specifically, the group with the lowest probability for *'others'* across all frames is selected as the predicted group $\hat{k}$(Eq. (13)). The final result is the prediction (excluding class *'others'*) from the classifier corresponding to the identified group. During inference, temporal logit adjustment is not used, *i.e.* inference applies argmax on predicted probabilities $\hat{p}$ instead of adjusted probability $\tilde{p}$ through temporal logit adjustment.

$$\hat{k} = \arg\min_i \frac{1}{T} \sum_t \hat{p}(o_t^{(i)}), \quad \hat{y}_t = \arg \max_{c_t^{(\hat{k})} \neq \text{others}} \hat{p}(c_t^{(\hat{k})}), \tag{13}$$

where $c_t^{(\hat{k})}$ represent class c in group $\hat{k}$ at time t.

5 Experiments

5.1 Dataset, Implementation, and Evaluation

Dataset. We evaluate our method on five datasets: Breakfast Actions [26] YouTube Instructional Videos [1], Assembly101 [39], GTEA [16] and 50Salads [44]. We follow the same settings and splits in [14, 39]. We split classes in these datasets into Head and Tail according to their frequency, with distributions for Breakfast and YouTube shown in Fig. 2 and others in the Supplementary.

Table 1: Class split and imbalance of used datasets. The imbalance ratio is the number of samples in the most frequent class divided by the number in the least frequent class.

Dataset	Head group		Tail group		Imbalance ratio
	#actions	#frames	#actions	#frames	
Breakfast	20	$\geq 5\times10^4$	28	$\leq 5\times10^4$	639
YouTube	14	≥ 500	32	≤ 500	558
Assembly101	31	$\geq 1.8\times10^5$	171	$\leq 1.8\times10^5$	2604
50salads	6	$\geq 4\times10^4$	13	$\leq 4\times10^4$	6
GTEA	5	≥ 2000	6	≤ 2000	24

Implementation Details. We use a temporal convolution model, MSTCN [14], and a transformer model, ASFormer [50] as backbones. We retrain these models using the same protocols, settings, and pre-extracted I3D features as the original papers; see the Supplementary for details and hyper-parameters. During training, activity labels or clustering results serve as group labels. During testing, the group label is inferred from predictions of the class *'others'* in that group.

Evaluation metrics. Temporal segmentation is traditionally evaluated [14, 42, 49, 50] using Mean over Frames accuracy for frames, Edit distance and F1-score at various IoU thresholds $(0.10, 0.25, 0.50)$ for segments. However, these metrics aggregate globally across all samples, masking tail class performance. To assess each class effectively, we use balanced per-class recall for frame-wise and per-class F1@0.25 score for segment-wise evaluations, following long-tailed works [22, 45, 46]. The balanced metrics ensure that performance is not largely driven by the head classes. We compute the average performance within the head and tail groups and their harmonic mean to reflect the balance in our predictions accurately. See Suppl. for the F1 score at other thresholds and global metrics.

5.2 Benchmark Comparisons

We compare our work against seven long-tail methods across different datasets and backbones in Tab. 2 on YouTube & Tab. 3 on Breakfast. We report per-class accuracy and F1 scores at IoU threshold 25% for Head and Tail groups, respectively, and their harmonic mean (Hmean) to ensure that the models perform well

Table 2: Comparison on YouTube with harmonic mean on head and tail classes over three runs. (Global metrics in grey are for reference. See Suppl. for standard deviation).

Model	Type	Frame acc			Segment F1@25			Global	
		Head	Tail	Hmean	Head	Tail	Hmean	Acc	F1@25
AsFormer	-	53.1	17.2	26.0	47.6	20.2	28.4	69.8	45.6
+ CB [11]	reweight	-2.2	+2.9	+2.8	-0.9	+1.0	+0.8	-0.2	-0.2
+ Focal [32]	reweight	-2.4	+0.8	+0.1	-2.2	+2.0	+1.4	-0.1	**+1.0**
+ BAGS [31]	ensemble	-1.2	+3.2	+3.3	-0.8	+1.8	+1.6	-0.5	-0.5
+ τ-norm [22]	post-hoc	+1.2	+1.3	+1.6	-1.0	+0.3	+0.1	-0.8	-1.3
+ LA [35]	logit adj.	+0.8	+4.9	+5.3	-0.7	+2.3	+2.1	-1.9	-0.5
+ LDAM [7]	logit adj.	+1.2	+3.6	+3.4	-1.7	+1.4	+0.9	-0.8	-0.8
+ Seesaw [46]	logit adj.	-0.6	+1.9	+2.0	-1.9	+1.3	+0.8	-0.6	+0.1
+ G-TLA(ours)	logit adj.	**+2.3**	**+6.8**	**+7.5**	**-0.3**	**+5.1**	**+4.6**	**+0.1**	+0.6
MSTCN	-	46.0	15.5	23.2	39.0	16.8	23.5	68.0	39.1
+ CB [11]	reweight	-0.2	+2.5	+2.7	+1.2	-0.4	-0.2	-1.7	-0.4
+ Focal [32]	reweight	+1.7	+1.5	+1.8	+1.9	+1.2	+1.9	-0.5	+0.9
+ BAGS [31]	ensemble	-0.4	+2.1	+2.1	+2.5	+0.5	+0.9	-0.8	+1.0
+ τ-norm [22]	post-hoc	+0.5	+1.1	+1.3	0.0	-0.6	-0.6	-0.7	-1.1
+ LA [35]	logit adj.	0.0	+2.1	+2.3	+0.4	-0.8	-0.7	-1.0	-0.3
+ LDAM [7]	logit adj.	-2.3	+0.3	0.0	-2.1	+0.6	+0.2	-0.4	+0.1
+ Seesaw [46]	logit adj.	+1.8	+1.4	+1.8	+1.8	+0.3	+0.6	**-0.1**	+0.6
+ G-TLA(ours)	logit adj.	**+2.7**	**+6.3**	**+6.8**	**+2.7**	**+3.3**	**+3.6**	-0.4	**+1.1**

across all classes, not just the head or tail. Some methods, like CB [11], enhance tail group classes at the expense of head group classes (see Tab. 2), while others, like Focal [32], primarily enhance head classes with little improvement for tail classes (see Tab. 3). Furthermore, some methods result in drops in segment-wise metric, $F1@25$, despite improved frame accuracy, indicating over-segmentation. For example, LA [35] and Seesaw [46] in Tab. 2 sacrifice head class F1 scores despite achieving promising frame accuracy gains. We also observe that ensemble methods generally perform best across the long-tail method types, followed by LA, reweighting, and post-hoc methods.

Our method, G-TLA, demonstrates strong performance across different class groups, metrics, datasets, and backbones, effectively addressing the long-tail problem while minimizing over-segmentation. Using the Asformer backbone, our method outperforms the next best model by 2.2% on YouTube and 1.5% on Breakfast on frame accuracy, and by 2.5% and 1.0% on F1@25, respectively. When using the MSTCN backbone, we outperform the next best model by 4.1% on YouTube and 2.6% on Breakfast on frame accuracy, and by 1.7% and 4.1% on F1@25, respectively. Our method not only enhances frame accuracy but also consistently improves F1@25. Our method is further tested with the SOTA DiffAct [33] backbone on Breakfast(Tab. 4) to verify its effectiveness, improving per-class performance without compromising overall performance. In summary, we find that directly applying standard long-tail models to temporal segmentation yields subpar performance, underscoring the significance of our method specifically designed to address the challenges posed by procedural activities.

In Tab. 5, we present our model's performance on a recent long-tailed dataset Assembly101 [39], along with two commonly used datasets, GTEA [16] and 50Salads [44], which have smaller vocabulary sizes and are less imbalanced. For

Table 3: Comparison on Breakfast with harmonic mean on head and tail classes over three runs. (Global metrics in grey are for reference. See Suppl. for standard deviation).

Model	Type	Frame acc			Segment F1@25			Global	
		Head	Tail	Hmean	Head	Tail	Hmean	Acc	F1@25
AsFormer	-	69.7	39.8	50.7	69.9	43.9	53.9	72.4	69.9
+ CB [11]	reweight	+0.5	+1.0	+0.9	+1.3	+0.8	+1.0	-0.5	-0.2
+ Focal [32]	reweight	+0.2	-0.7	-0.5	+1.4	-0.2	+1.2	-0.1	+0.5
+ BAGS [31]	ensemble	-0.2	+0.8	+0.5	+1.6	+0.5	+1.0	-0.6	-1.0
+ τ-norm [22]	post-hoc	-0.1	+1.3	+0.8	0.0	+0.4	+0.3	-0.2	-0.8
+ LA [35]	logit adj.	+0.4	+0.6	+0.6	+0.3	+0.8	+1.0	+0.1	-0.2
+ LDAM [7]	logit adj.	+0.4	+1.2	+1.0	+0.3	+1.1	+1.2	**+0.2**	+0.7
+ Seesaw [46]	logit adj.	+0.4	+1.3	+1.1	+0.5	+1.5	+1.6	+0.1	+0.2
+ G-TLA(ours)	logit adj.	**+0.6**	**+3.4**	**+2.6**	**+1.8**	**+2.6**	**+2.6**	-0.2	**+1.0**
MSTCN	-	65.1	37.7	47.7	53.3	38.7	44.8	67.7	57.9
+ CB [11]	reweight	-1.0	+1.6	+1.1	+0.8	+0.7	+0.8	-0.3	0.0
+ Focal [32]	reweight	+1.0	-1.6	-1.0	+0.3	-0.7	-0.4	+0.8	-0.4
+ BAGS [31]	ensemble	+0.9	+1.8	+1.7	+4.4	+1.5	+2.6	+0.8	+1.9
+ τ-norm [22]	post-hoc	+0.2	-1.5	-1.1	-0.6	-1.3	-1.0	+0.2	-0.9
+ LA [35]	logit adj.	-0.7	+2.4	+2.1	+2.7	0.0	+0.9	-0.1	0.0
+ LDAM [7]	logit adj.	+0.7	+0.1	+0.3	+0.7	+0.8	+0.8	-0.2	+0.2
+ Seesaw [46]	logit adj.	+1.2	+2.5	+2.4	+0.7	+0.3	+0.5	+0.9	-0.1
+ G-TLA(ours)	logit adj.	**+2.5**	**+5.3**	**+5.0**	**+6.8**	**+6.5**	**+6.7**	**+2.6**	**+5.0**

Table 4: Additional results with DiffAct on Breakfast.

Model	Frame acc			Segment F1@25			Global	
	Head	Tail	Hmean	Head	Tail	Hmean	Acc	F1@25
DiffAct	74.8	42.6	54.3	77.5	49.2	60.2	76.5	75.3
+ CB [11]	74.1	43.3	54.7	77.5	49.1	60.2	75.7	75.8
+ LA [35]	75.3	44.5	55.9	78.1	49.0	60.2	76.3	75.6
+ Seesaw [46]	73.2	38.9	50.8	76.0	44.0	55.8	73.6	72.0
+ G-TLA(ours)	**75.6**	**45.5**	**56.8**	**78.4**	**50.4**	**61.4**	**76.6**	**76.1**

GTEA and Assembly101, we apply clustering to determine the group for each sequence (3 groups for GTEA and 2 groups for Assembly101), with each group containing several activities. For 50Salads, which contains a single activity, we forgo the group-wise framework and exclusively implement temporal logit adjustment. Our method demonstrates competitive results over the baseline by a large margin, further validating its effectiveness. Notably, the tail performance is even higher than the head for GTEA, emphasizing the lack of imbalance.

5.3 Ablation Studies

We present ablations on Breakfast; please see the Suppl. for other datasets.

G-TLA Components. We assess the contributions of group-wise classification (GP), naive logit adjustment (LA), and temporal factors (TF) in Tab. 6. GP significantly reduces over-segmentation by 5.5% with MSTCN and 1.1% for AS-Former. Naive LA improves tail accuracy by 2.4% but decreases head accuracy by 0.7% for MSTCN. Temporal priors help reduce false positives that violate the ordering prior, further improving LA. Combining all our components within G-TLA achieves a balanced result in both frame and segment metrics.

Table 5: Additional results on 50salads, GTEA, and Assembly101.

Dataset	Model	Frame acc			Segment F1@25			Global	
		Head	Tail	Hmean	Head	Tail	Hmean	Acc	F1@25
50salads	**AsFormer**	90.6	77.4	83.5	87.5	80.3	83.8	85.2	82.3
	+ G-TLA(ours)	**90.8**	**79.7**	**84.9**	**89.4**	**83.1**	**86.1**	**86.3**	**84.6**
	MSTCN	87.7	70.0	77.9	85.7	72.1	78.3	81.1	75.9
	+ G-TLA(ours)	**89.0**	**71.7**	**79.4**	**86.8**	**73.8**	**79.8**	**81.9**	**77.3**
GTEA	**AsFormer**	**80.6**	81.7	81.2	**72.5**	85.4	78.4	81.1	89.4
	+ G-TLA(ours)	80.2	**84.5**	**82.3**	72.0	**90.4**	**80.2**	**81.2**	**89.6**
	MSTCN	**77.6**	80.3	78.9	69.4	86.6	77.0	78.0	87.2
	+ G-TLA(ours)	77.5	**83.7**	**80.5**	**69.5**	**90.3**	**78.5**	**78.6**	**87.9**
Assembly101	**AsFormer**	35.2	5.7	9.8	29.0	4.8	8.2	**41.1**	**30.4**
	+ G-TLA(ours)	**36.8**	**9.2**	**14.7**	**30.7**	**8.3**	**13.1**	41.0	29.8
	MSTCN	33.9	4.7	8.2	26.3	3.9	6.8	**39.8**	27.2
	+ G-TLA(ours)	**34.9**	**8.0**	**13.0**	**30.2**	**5.8**	**9.7**	39.2	**28.5**

Table 6: Ablate group classification(GP), logit adjustment(LA), temporal factor(TF).

GP	LA	TF	MSTCN						ASFormer					
			Frame acc			Seg. F1			Frame acc			Seg. F1		
			Head	Tail	Hmean	Head	Tail	Hmean	Head	Tail	Hmean	Head	Tail	Hmean
✗	✗	✗	65.1	37.7	47.7	53.3	38.7	44.8	69.7	39.8	50.7	69.9	43.9	53.9
✗	✓	✗	64.4	40.1	49.8	56.0	38.7	45.7	70.1	40.4	51.3	71.2	44.7	54.9
✗	✓	✓	65.7	41.3	50.7	56.3	38.9	46.0	69.4	41.7	52.2	70.7	44.8	54.9
✓	✗	✗	67.5	40.8	50.9	**60.3**	44.8	51.3	69.7	40.8	51.5	71.1	44.9	55.0
✓	✓	✗	66.5	41.8	51.3	59.9	44.7	51.2	70.2	41.4	52.1	71.4	44.7	55.0
✓	✓	✓	**67.6**	**43.0**	**52.7**	60.1	**45.2**	**51.5**	**70.3**	**43.2**	**53.3**	**71.7**	**46.5**	**56.5**

η **for group-wise classification.** Tab. 7 shows the impact of hyperparameter η in Eq. (12). A small η reduces suppression of tail classes by down-weighting negative gradients from the *'others'* class, but if too small, it harms group identification during inference. Conversely, a large η over-emphasizes the *'others'* class, harming tail performance. The results suggest an optimal value of η is 0.5.
τ **for temporal logit adjustment.** A small τ in Eq. (9) represents minimal adjustment, resulting in less improvement for tail classes in Tab. 8. Conversely, a large τ biases towards tail classes and introduces more false positives, negatively impacting head and segment-wise performance. Our experiments show that $\tau = 0.5$ achieves optimal performance.

5.4 Analyzing the effects of G-TLA

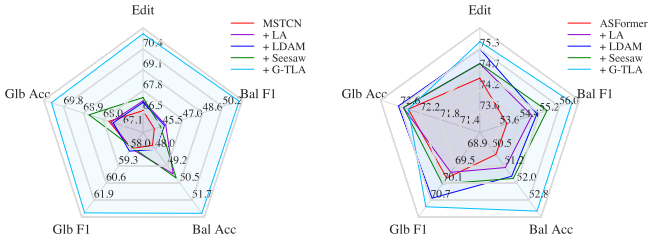
Individual metrics. Fig. 5 visualizes various methods' global and per-class results with radar charts, including global accuracy, F1 score, and Edit score, as well as the harmonic mean of balanced accuracy and F1@25 score. Our method is significantly more balanced, as indicated by the largest enclosed area. In particular, our method excels in segment-wise performance, including Edit score and global & balanced F1 score, demonstrating our approach's effectiveness in reducing over-segmentation while enhancing balanced accuracy.
Trade-off trends. Long-tail temporal action segmentation exhibits two distinct trade-offs. First is a head-tail trade-off, where the head is impacted negatively

Table 7: Varying η for group-wise classification, with fixed number of groups $n = 10$ and $\tau = 0.5$ on MSTCN

η	Frame acc			Seg. F1		
	Head	Tail	Hmean	Head	Tail	Hmean
0.1	67.3	41.0	51.0	60.4	44.0	50.9
0.3	67.2	42.1	51.8	60.5	44.8	51.5
0.5	**67.6**	**43.0**	**52.7**	60.1	45.2	51.5
0.7	67.5	42.3	52.1	**61.1**	**45.7**	**52.2**

Table 8: Varying τ for temporal logit adjustment, with fixed number of groups $n = 10$ and $\eta = 0.5$ on MSTCN

τ	Frame acc			Seg. F1		
	Head	Tail	Hmean	Head	Tail	Hmean
0.1	67.4	40.8	50.9	60.9	43.3	50.6
0.3	67.5	41.8	51.6	**61.0**	43.2	50.6
0.5	**67.6**	**43.0**	**52.7**	60.1	**45.2**	**51.5**
0.7	**67.6**	42.4	52.1	60.6	43.9	50.9



(a) MSTCN. **(b)** AsFormer.

Fig. 5: Radar charts of different logit adjustment methods, measuring the performance along balanced and global metrics on Breakfast with MSTCN and AsFormer.

when the tail is improved. Second is a frame-segment trade-off, where enhancing the tail improves frame metrics but hurts segment metrics. The extent of these trade-offs is directly influenced by the hyperparameters of respective long-tail methods, as shown in Fig. 6. Head performance involves head group metrics of per-class accuracy and F1 score, and tail performance involves tail group metrics. Frame performance measures the average of global and per-class accuracy, while segment performance involves Edit score, global F1@25, and per-class F1@25. As the plot indicates, our method achieves a much better balance than others. The curve of our method consistently remains above others, emphasizing the boost of tail classes. This contributes to improved balanced metrics while maintaining competitive head and segment performance. See the Supplementary for hyperparameters and trends on other datasets and backbones.

Qualitative & quantitative results. We qualitatively compare the predictions of logit adjustment methods. Fig. 7 shows the output of MSTCN on Breakfast and YouTube, revealing the common issue of introducing activity-irrelevant classes when enhancing tail classes. For instance, emphasizing the tail class *'stir milk'* on Breakfast introduces false positives in unrelated activity *"making cereal"*. Additionally, existing methods ignore action ordering, resulting in false positives that violate temporal priors, *e.g.* on Youtube, action *'UNSCREW WHEEL'* is wrongly predicted before *'START LOOSE'* for activity *"changing tire"*. Our proposed method effectively addresses both types of false positives, enhancing the overall prediction logic. We also report the quantitative results on Breakfast with MSTCN in Fig. 8. Our method reduces

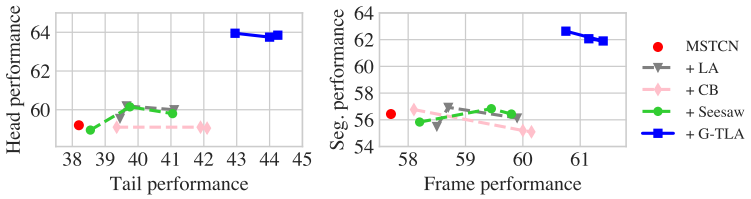
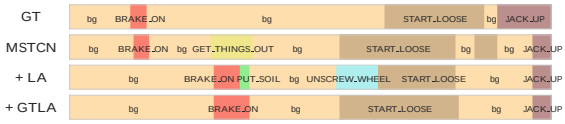


Fig. 6: Head-Tail & Frame-Segment trade-offs on Breakfast with MSTCN.

(a) *"making cereals"* activity on Breakfast using MSTCN.



(b) *"changing tire"* activity on Youtube using MSTCN.

Fig. 7: G-TLA effectively reduces activity-irrelevant predictions, *e.g.* actions *'spoon powder'* and *'stir milk'* on Breakfast, and *'PUT SOIL'* on Youtube. G-TLA also mitigates predictions that violate temporal priors, *e.g.* *'UNSCREEW WHEEL'* occurring before *'START LOOSE'* on Youtube.

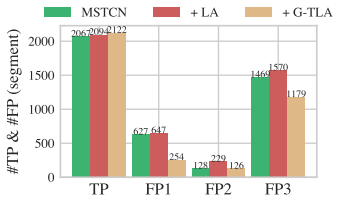


Fig. 8: Distribution of True (TP) and False Positives (FP), categorized into: FP1 (activity-irrelevant), FP2 (activity-relevant with ordering violations), and FP3 (activity-relevant and following action ordering derived from temporal bounds).

the number of segments for activity-irrelevant(FP1) and ordering-violated(FP2) false positives, mitigating over-segmentation. Meanwhile, it also helps to increase true positives(TP) and reduce false positives that are both activity-relevant and ordering-valid(FP3), mainly due to the group-wise classification.

Group identification accuracy. Group identification is crucial and impacts the final performance. Our group-wise classification improves activity identification. For instance, we achieve 90.1% accuracy compared to 87.2% of the MSTCN baseline on Breakfast. More details and results are in the Supplementary.

6 Conclusion

This paper targets long-tail temporal action segmentation, addressing challenges from temporal class correlations and performance trade-offs of head & tail classes at frame- & segment-level. Our proposed Group-wise Temporal Logit Adjustment (G-TLA) scheme integrates video activity labels and action order priors to capture class inter-dependencies, enhancing balanced performance while maintaining global performance. Future efforts would prioritize highly imbalanced datasets, considering both long-tail and few-shot learning scenarios.

Acknowledgements

This research is supported by the Ministry of Education, Singapore, under the Academic Research Fund Tier 1 (FY2022).

References

1. Alayrac, J.B., Bojanowski, P., Agrawal, N., Sivic, J., Laptev, I., Lacoste-Julien, S.: Unsupervised learning from narrated instruction videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4575–4583 (2016)
2. Bhattacharya, S., Kalayeh, M.M., Sukthankar, R., Shah, M.: Recognition of complex events: Exploiting temporal dynamics between underlying concepts. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2235–2242 (2014)
3. Brodersen, K.H., Ong, C.S., Stephan, K.E., Buhmann, J.M.: The balanced accuracy and its posterior distribution. In: 2010 20th international conference on pattern recognition. pp. 3121–3124. IEEE (2010)
4. Buda, M., Maki, A., Mazurowski, M.A.: A systematic study of the class imbalance problem in convolutional neural networks. Neural networks **106**, 249–259 (2018)
5. Byrd, J., Lipton, Z.: What is the effect of importance weighting in deep learning? In: International conference on machine learning. pp. 872–881. PMLR (2019)
6. Cai, J., Wang, Y., Hwang, J.N.: Ace: Ally complementary experts for solving long-tailed recognition in one-shot. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 112–121 (2021)
7. Cao, K., Wei, C., Gaidon, A., Arechiga, N., Ma, T.: Learning imbalanced datasets with label-distribution-aware margin loss. Advances in neural information processing systems **32** (2019)
8. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6299–6308 (2017)
9. Cheng, Y., Fan, Q., Pankanti, S., Choudhary, A.: Temporal sequence modeling for video event detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2227–2234 (2014)
10. Collell, G., Prelec, D., Patil, K.: Reviving threshold-moving: a simple plug-in bagging ensemble for binary and multiclass imbalanced data. arXiv preprint arXiv:1606.08698 (2016)
11. Cui, Y., Jia, M., Lin, T.Y., Song, Y., Belongie, S.: Class-balanced loss based on effective number of samples. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9268–9277 (2019)
12. Ding, G., Sener, F., Yao, A.: Temporal action segmentation: An analysis of modern techniques. IEEE Transactions on Pattern Analysis and Machine Intelligence (2023)
13. Drummond, C., Holte, R.C., et al.: C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In: Workshop on learning from imbalanced datasets II. vol. 11, pp. 1–8 (2003)
14. Farha, Y.A., Gall, J.: Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3575–3584 (2019)

15. Fathi, A., Farhadi, A., Rehg, J.M.: Understanding egocentric activities. In: 2011 international conference on computer vision. pp. 407–414. IEEE (2011)
16. Fathi, A., Ren, X., Rehg, J.M.: Learning to recognize objects in egocentric activities. In: CVPR 2011. pp. 3281–3288. IEEE (2011)
17. Gao, S.H., Han, Q., Li, Z.Y., Peng, P., Wang, L., Cheng, M.M.: Global2local: Efficient structure search for video action segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16805–16814 (2021)
18. Han, H., Wang, W.Y., Mao, B.H.: Borderline-smote: a new over-sampling method in imbalanced data sets learning. In: Advances in Intelligent Computing: International Conference on Intelligent Computing, ICIC 2005, Hefei, China, August 23-26, 2005, Proceedings, Part I 1. pp. 878–887. Springer (2005)
19. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Transactions on knowledge and data engineering **21**(9), 1263–1284 (2009)
20. Hu, X., Jiang, Y., Tang, K., Chen, J., Miao, C., Zhang, H.: Learning to segment the tail. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 14045–14054 (2020)
21. Huang, C., Li, Y., Loy, C.C., Tang, X.: Learning deep representation for imbalanced classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5375–5384 (2016)
22. Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., Kalantidis, Y.: Decoupling representation and classifier for long-tailed recognition. arXiv preprint arXiv:1910.09217 (2019)
23. Kim, B., Kim, J.: Adjusting decision boundary for class imbalanced learning. IEEE Access **8**, 81674–81685 (2020)
24. King, G., Zeng, L.: Logistic regression in rare events data. Political analysis **9**(2), 137–163 (2001)
25. Kuehne, H., Arslan, A., Serre, T.: The language of actions: Recovering the syntax and semantics of goal-directed human activities. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 780–787 (2014)
26. Kuehne, H., Arslan, A., Serre, T.: The language of actions: Recovering the syntax and semantics of goal-directed human activities. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 780–787 (2014)
27. Kuehne, H., Richard, A., Gall, J.: A hybrid rnn-hmm approach for weakly supervised temporal action segmentation. IEEE transactions on pattern analysis and machine intelligence **42**(4), 765–779 (2018)
28. Lea, C., Flynn, M.D., Vidal, R., Reiter, A., Hager, G.D.: Temporal convolutional networks for action segmentation and detection. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 156–165 (2017)
29. Lei, P., Todorovic, S.: Temporal deformable residual networks for action segmentation in videos. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6742–6751 (2018)
30. Li, S.J., AbuFarha, Y., Liu, Y., Cheng, M.M., Gall, J.: Ms-tcn++: Multi-stage temporal convolutional network for action segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1–1 (2020). https://doi.org/10.1109/TPAMI.2020.3021756
31. Li, Y., Wang, T., Kang, B., Tang, S., Wang, C., Li, J., Feng, J.: Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10991–11000 (2020)

32. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
33. Liu, D., Li, Q., Dinh, A.D., Jiang, T., Shah, M., Xu, C.: Diffusion action segmentation. In: International Conference on Computer Vision (ICCV) (2023)
34. Menon, A., Narasimhan, H., Agarwal, S., Chawla, S.: On the statistical consistency of algorithms for binary classification under class imbalance. In: International Conference on Machine Learning. pp. 603–611. PMLR (2013)
35. Menon, A.K., Jayasumana, S., Rawat, A.S., Jain, H., Veit, A., Kumar, S.: Long-tail learning via logit adjustment. arXiv preprint arXiv:2007.07314 (2020)
36. Perrett, T., Damen, D.: Recurrent assistance: cross-dataset training of lstms on kitchen tasks. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 1354–1362 (2017)
37. Perrett, T., Sinha, S., Burghardt, T., Mirmehdi, M., Damen, D.: Use your head: Improving long-tail video recognition. arXiv preprint arXiv:2304.01143 (2023)
38. Ren, M., Zeng, W., Yang, B., Urtasun, R.: Learning to reweight examples for robust deep learning. In: International conference on machine learning. pp. 4334–4343. PMLR (2018)
39. Sener, F., Chatterjee, D., Shelepov, D., He, K., Singhania, D., Wang, R., Yao, A.: Assembly101: A large-scale multi-view video dataset for understanding procedural activities. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21096–21106 (2022)
40. Shen, L., Lin, Z., Huang, Q.: Relay backpropagation for effective learning of deep convolutional neural networks. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14. pp. 467–482. Springer (2016)
41. Singh, B., Marks, T.K., Jones, M., Tuzel, O., Shao, M.: A multi-stream bi-directional recurrent neural network for fine-grained action detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1961–1970 (2016)
42. Singhania, D., Rahaman, R., Yao, A.: Coarse to fine multi-resolution temporal convolutional network. arXiv preprint arXiv:2105.10859 (2021)
43. Stein, S., McKenna, S.J.: Combining embedded accelerometers with computer vision for recognizing food preparation activities. In: Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing. pp. 729–738 (2013)
44. Stein, S., McKenna, S.J.: Combining embedded accelerometers with computer vision for recognizing food preparation activities. In: Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing. pp. 729–738 (2013)
45. Tan, J., Wang, C., Li, B., Li, Q., Ouyang, W., Yin, C., Yan, J.: Equalization loss for long-tailed object recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11662–11671 (2020)
46. Wang, J., Zhang, W., Zang, Y., Cao, Y., Pang, J., Gong, T., Chen, K., Liu, Z., Loy, C.C., Lin, D.: Seesaw loss for long-tailed instance segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9695–9704 (2021)
47. Wang, T., Li, Y., Kang, B., Li, J., Liew, J., Tang, S., Hoi, S., Feng, J.: The devil is in classification: A simple framework for long-tail instance segmentation. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16. pp. 728–744. Springer (2020)

48. Wang, Y.X., Ramanan, D., Hebert, M.: Learning to model the tail. Advances in neural information processing systems **30** (2017)
49. Wang, Z., Gao, Z., Wang, L., Li, Z., Wu, G.: Boundary-aware cascade networks for temporal action segmentation. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16. pp. 34–51. Springer (2020)
50. Yi, F., Wen, H., Jiang, T.: Asformer: Transformer for action segmentation. arXiv preprint arXiv:2110.08568 (2021)
51. Zhang, J., Liu, L., Wang, P., Shen, C.: To balance or not to balance: A simple-yet-effective approach for learning with long-tailed distributions. arXiv preprint arXiv:1912.04486 (2019)
52. Zhang, X., Wu, Z., Weng, Z., Fu, H., Chen, J., Jiang, Y.G., Davis, L.S.: Videolt: large-scale long-tailed video recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7960–7969 (2021)
53. Zhao, Y., Chen, W., Tan, X., Huang, K., Zhu, J.: Adaptive logit adjustment loss for long-tailed visual recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 3472–3480 (2022)
54. Zhou, B., Cui, Q., Wei, X.S., Chen, Z.M.: Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9719–9728 (2020)