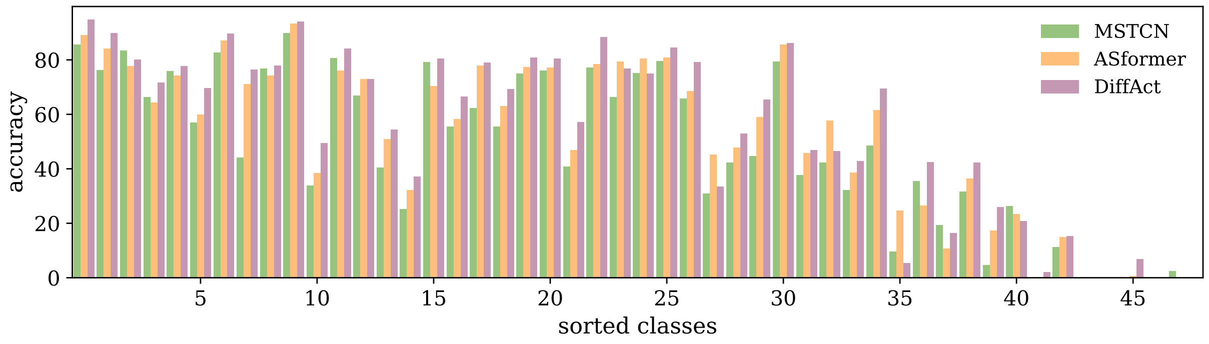
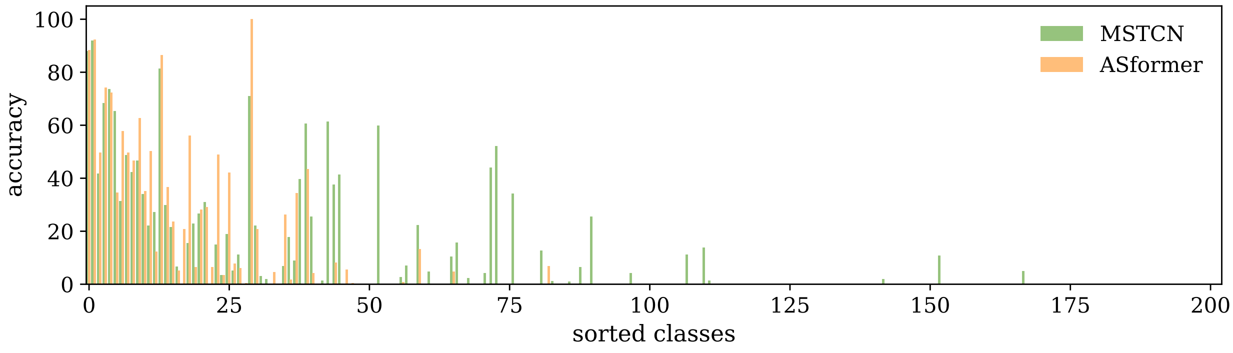
Supplementary

A. Long-tail Problem in Temporal Action Segmentation

Temporal action segmentation methods [13, 31, 52] often ignore the long-tail problem, leading to poor performance on tail classes. For instance, state-of-the-art models like MSTCN [13], ASFormer [52], and DiffAct [31] fail to predict tail classes accurately. On Breakfast dataset, MSTCN and ASFormer each have zero accuracy for 5 out of 48 classes, while DiffAct misses 4 classes entirely. On Assembly101 dataet, there are 30 classes do not appear in test set. Except those non-appeared classes, MSTCN and ASFormer achieve zero accuracy for 106 and 128 classes of 141 tail classes respectively. Details can be seen in Fig. 5.



(a) Breakfast



(b) Assembly101

Figure 5: Class-wise accuracy distribution.

B. Convert Optimization to Weighted Cross-entropy

Proposition 1. *Given a timestamp t and its previous action u_t, the optimal classifier of*

$$\max_f \sum_{i,j,k} G_{i,j,k} C_{i,j,k}[f]$$

for a gain matrix $G \in \mathbb{R}^{L \times L \times L+1}$ and the t^{th} frame takes the form:

$$f^*(X, u_t) \in \arg\max_{j \in [L]} \sum_i p_i(X) G_{i,j,u_t}$$

where $p_i(X)$ is the estimated conditional probability for class i at the current frame t by classifier f.

Proof.

$$\begin{aligned}\sum_{i,j,k} G_{i,j,k}C_{i,j,k}[f] &= \mathbb{E}_{(X,y_t,u_t)}[\sum_{i,j,k} G_{i,j,k}\mathbb{1}(y_t = i, \hat{y}_t = j, u_t = k)] \\ &= \mathbb{E}_{(X,y_t,u_t)}[\sum_{j} G_{y_t,j,u_t}\mathbb{1}(\hat{y}_t = j)] \\ &= \mathbb{E}_{(X,u_t)}\mathbb{E}_{(y_t|X,u_t)}[\sum_{j} G_{y_t,j,u_t}\mathbb{1}(\hat{y}_t = j)] \\ &= \mathbb{E}_{(X,u_t)}[\sum_{i,j} p_i(X)G_{i,j,u_t}\mathbb{1}(\hat{y}_t = j)]\end{aligned}$$

We use the fact in frame-wise classification where the prediction for y_t does not depend on u_t. It suffices to maximize the above objective point-wise to compute the Bayes-optimal classifier. To predict for a frame labelled as y_t of given input X and the label for the previous action u_t, the prediction should maximize the term in the expectation.

$$f^*(X, u_t) \in \underset{j\in[L]}{\arg\max} \sum_i p_i(X)G_{i,j,u_t}$$

where $G_{:,:,u_t}$ is a matrix, denoting a slice of G. □

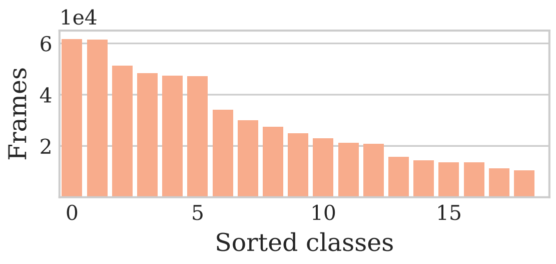
In our case, the gain matrix $G_{:,:,u_t}$ is diagonal. The optimal classifier takes the form

$$f^*(X, u_t) \in \underset{i\in[L]}{\arg\max}\, p_i(X)G_{i,i,u_t} \propto \underset{i\in[L]}{\arg\min} -G_{i,i,u_t} \log p_i(X)$$

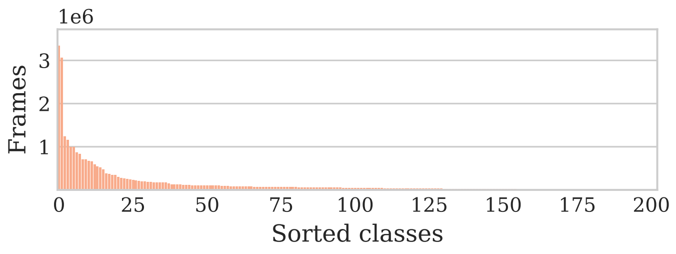
which is the reweighted cross entropy loss and is calibrated for the diagonal gain matrix.

C. Experimental Setting

Dataset distribution. Extra data distribution of 50salads and Assembly101 is illustrated in Fig. 6.



(a) 50salads



(b) Assembly101

Figure 6: Data distribution of 50salads and Assembly101.

Hyperparameters: The used hyperparameters for each dataset, method, and backbone are shown in Table 7. We omit τ-norm [21] as the results always favour $\tau = 1.0$ for τ-norm. In our method, we fix the hyperparameter ε in Eq. (5) as 0.9, and the learning rate for multiplier γ in Algorithm 1 as 0.01.

Table 7: Hyperparameters summary

Data	Model	**Focal** [30]	**CB** [9]	**LA** [32]	**CSL**(ours)
		γ	β	τ	τ
Breakfast	MSTCN	0.5	0.9	0.5	0.5
	AsFormer	1.5	0.9	0.1	0.3
	DiffAct	-	0.99	0.3	0.7
50salads	MSTCN	1.5	0.9	0.5	0.7
	AsFormer	0.5	0.99	0.3	0.9
Assembly	MSTCN	0.5	0.9	0.1	0.3
	AsFormer	0.5	0.9	0.3	0.1

D. Additional Results

Global performance. Evaluation in the main paper primarily focuses on per-class performance, as it better reflects the extent to which the long-tail problem is addressed. Since existing works in temporal action segmentation commonly report global performance, we also present detailed global results across different datasets, backbones, and methods in Table 8 for completeness. Notably, our method, which includes constraints for detecting transitions, demonstrates large improvements on global segment-wise metrics, *i.e.*, F1 and edit scores. Although our method may not always lead in frame-wise performance, it still delivers competitive results. Balancing global and balanced results is challenging due to the trade-off: improving tail often boosts per-class results at the expense of head performance, resulting in the drop in global results. Our method achieves a good trade-off by significantly enhancing per-class performance while still showing competitive results on global metrics.

Table 8: Result summary on global metrics.

Model	**Breakfast**					**50salads**					**Assembly101**				
	F1@{10,25,50}			Edit	Acc.	F1@{10,25,50}			Edit	Acc.	F1@{10,25,50}			Edit	Acc.
MSTCN	63.2	57.9	46.0	66.6	67.7	78.5	75.9	67.0	71.4	81.1	30.8	27.2	20.5	30.1	**39.8**
+ CB [9]	63.6	57.9	45.7	66.8	67.4	77.7	75.5	65.8	71.1	81.0	30.0	26.7	20.2	28.4	39.7
+ LA [32]	63.1	57.9	45.6	67.2	67.6	78.2	75.2	66.9	70.4	80.8	29.4	26.1	20.0	29.2	39.2
+ Focal [30]	63.1	57.5	45.5	67.3	**68.5**	78.8	76.1	67.6	70.8	81.7	30.6	27.0	20.0	30.7	39.2
+ τ-norm [21]	62.4	57.0	45.1	66.3	67.9	77.7	75.3	66.5	70.8	81.1	31.1	27.4	20.7	30.5	39.6
+ ours(S-NCM)	**69.3**	**64.0**	**50.9**	**67.7**	67.5	**81.3**	**79.0**	**70.2**	**74.0**	**81.8**	**32.9**	**29.5**	**22.8**	**30.8**	39.1
ASFormer	75.5	69.9	56.1	74.5	**72.4**	84.8	82.3	75.1	79.0	85.2	34.4	30.4	21.5	31.8	41.1
+ CB [9]	75.6	69.7	55.8	74.9	71.9	84.9	83.2	75.7	78.7	85.8	32.6	28.2	20.1	30.6	41.0
+ LA [32]	75.6	69.7	56.3	74.9	**72.4**	84.9	83.2	76.3	78.3	85.3	32.3	28.5	20.9	30.2	**41.3**
+ Focal [30]	**75.7**	**70.4**	56.2	**75.2**	72.3	85.7	83.5	75.7	79.6	84.6	34.1	30.3	22.4	32.1	41.2
+ τ-norm [21]	74.9	69.1	55.7	73.6	72.2	84.7	82.2	75.2	78.9	85.2	26.4	22.7	15.9	24.3	38.5
+ ours(S-NCM)	75.3	**70.4**	**57.5**	74.3	72.1	**86.0**	**84.0**	**77.8**	**80.3**	**86.0**	**34.8**	**31.7**	**23.8**	**32.9**	40.8

Transition detection. The transition constraints help focus on learning hard transitions. Fig. 7 presents the distribution of transition accuracy, as defined in Eq. (3) on Breakfast test set. Transitions are sorted according to the baseline performance. The results indicate that the model trained under the defined the constraints can detect more transitions, particularly in the tail section where the baseline model struggles to recognise them, demonstrating the efficacy of our transition constraints. Specifically, the baseline Asformer can detect 132 out of 167 transitions, while our cost-sensitive learning(CSL) method successfully detects 11 more transitions. Besides, our method achieves higher average transition accuracy 56.1% than the baseline 54.3%.

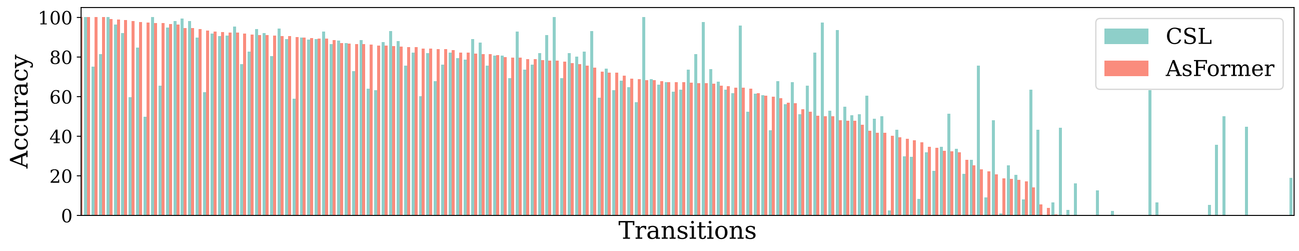


Figure 7: Transition accuracy for AsFormer on Breakfast testset.