# Reducing the Label Bias for Timestamp Supervised Temporal Action Segmentation

Kaiyuan Liu[1*] Yunheng Li[1*] Shenglan Liu[1†] Chenwei Tan[1] Zihang Shao[1]
[1]School of Computer Science, Dalian University of Technology, China
{1154864382,liyunheng,liusl,tcw2000,1317588161}@mail.dlut.edu.cn

In Sec. A, we consider the social impact of our work. The training efficiency of different attention architectures and the full architecture of our segmentation model is described in Sec. B. In Sec. C, we evaluate the number of masked frames of MTP and the order of Naive and MTP. Besides, we compare pseudo-labels of our initialized model and unsupervised method and study the impact of the weight of segmental confidence loss and the effect of different loss functions. Finally, we perform our D-TSTAS on rare annotation cases. In Sec. D, we validate the qualitative results of the CTE and the D-TSTAS.

## A. Social impact

Our D-TSTAS achieves competitive performance compared with the fully supervised setup by utilizing a single annotated frame of each action segment, which facilitates the development of action understanding tasks. The proposed approach can be applied to recognize frame-wise frames of complex actions in long untrimmed videos, e.g. instructional video analysis and surveillance applications. Meanwhile, privacy protection should be considered in practical applications.

## B. The segmentation model

**B.1 The training efficiency of different attention architectures.** We have explored different backbones for TSTAS, including TCN backbone in [2], local-attention transformer backbone in ASFormer [5], and linear-attention transformer backbone in Flowformer [4]. When batch size is larger than 1, the performance of the local-attention transformer on full-supervised TAS task degrades, as evident from issue#5 and issue#9 on the official repository. As shown in Tab. B.1, a similar degradation of performance is observed for the TSTAS task The model achieves better performance when the batch size is set to 1 rather than 8 but requires 3 times the training time on the smallest dataset like GTEA, which only contains 58 videos. When dealing with large-scale datasets such as Breakfast which has 1712 videos, the training time becomes even more unacceptable.

| Backbone | bz | Training Time(min) | F1@{10,25,50} | | | Edit | Acc |
|---|---|---|---|---|---|---|---|
| Local-attention [5] | 1 | 41 | 84.7 | 79.4 | 59.7 | 79.6 | 64.2 |
| | 8 | 13 | 42.8 | 38.7 | 26.7 | 33.1 | 40.5 |
| Linear-attention [4] | 8 | 8 | 76.0 | 71.4 | 55.3 | 71.7 | 69.0 |

Table B.1. Comparison with the different backbones under the same timestamp-supervised setting with Li et al. [2] on the GTEA dataset.

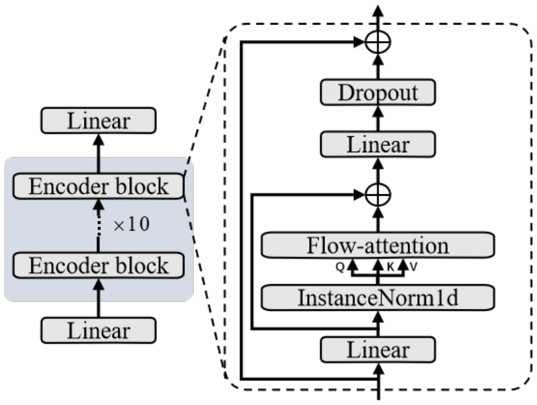

Figure B.1. The overview of our encoder block.

To balance training time and performance, we use Linear-attention [4] in our transformer backbone, which is beneficial for modeling long sequences.

**B.2 The architecture of the segmentation model.** In this section, we present the details of the structure of our segmentation model. We use the same encoder-decoder structured transformer as ASFormer [5], whose encoder consists of 10 encoder blocks. As shown in Fig. B.1, the Linear layer is used as the feed-forward layer, and the output of the InstanceNorm layer is applied as the inputs of the attention layer (Q, K, and V come from the same inputs). Instead of using local-attention, the flow-attention proposed by [4] with linear-complexity is used. Moreover, we utilize the hierarchical pattern [5] for the attention layers in both the encoder and decoder. Unlike the encoder, the Q and K in the decoder are obtained from the output of the InstanceNorm layer, while the V is obtained from the output of the previous layer $F_h$.

As shown in Fig. B.2, we describe the overall architec-

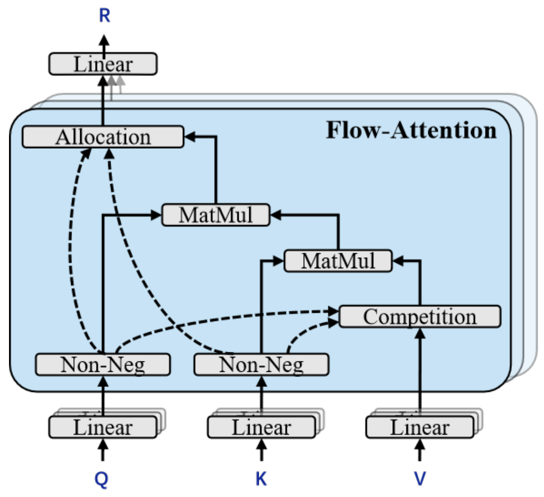

Figure B.2. The architecture of Flow-Attention [4]. Q denotes the queries, K is the keys, and V is the values.

| Number | Initialization | | CTE | F1@{10, 25, 50} | | | Edit | Acc |
|---|---|---|---|---|---|---|---|---|
| | MTP | Naive | | | | | | |
| | ✓ | | | 59.8 | 55.1 | 44.9 | 52.1 | 70.1 |
| 1(0.17%) | ✓ | ✓ | | 60.2 | 55.7 | 45.5 | 52.8 | 70.5 |
| | ✓ | ✓ | ✓ | 84.4 | 82.2 | 71.2 | 77.5 | 80.2 |
| | ✓ | | | 54.6 | 50.4 | 39.9 | 45.6 | 69.6 |
| 3(0.52%) | ✓ | ✓ | | 58.2 | 54.7 | 44.3 | 49.7 | 73.1 |
| | ✓ | ✓ | ✓ | 84.1 | 82.3 | 71.3 | 77.6 | 79.8 |
| | ✓ | | | 41.8 | 35.8 | 25.7 | 34.1 | 59.4 |
| 5(0.86%) | ✓ | ✓ | | 54.7 | 50.3 | 40.2 | 46.0 | 71.3 |
| | ✓ | ✓ | ✓ | 83.4 | 81.1 | 68.4 | 76.1 | 78.7 |
| | ✓ | | | 39.1 | 33.4 | 22.4 | 33.1 | 56.5 |
| 7(1.21%) | ✓ | ✓ | | 54.0 | 49.0 | 38.9 | 44.0 | 71.2 |
| | ✓ | ✓ | ✓ | 83.5 | 81.4 | 69.8 | 77.4 | 78.4 |
| | ✓ | | | 30.7 | 25.1 | 15.7 | 27.0 | 45.9 |
| 13(2.25%) | ✓ | ✓ | | 53.7 | 49.2 | 38.4 | 44.2 | 71.1 |
| | ✓ | ✓ | ✓ | 81.7 | 79.7 | 66.3 | 74.9 | 77.1 |
| | ✓ | | | 31.1 | 24.7 | 14.3 | 27.3 | 43.5 |
| 15(2.60%) | ✓ | ✓ | | 54.3 | 50.5 | 40.2 | 44.3 | 71.1 |
| | ✓ | ✓ | ✓ | 81.7 | 79.3 | 66.6 | 74.3 | 76.8 |
| | ✓ | | | 22.7 | 17.7 | 9.4 | 21.7 | 34.7 |
| 29(5.00%) | ✓ | ✓ | | 55.3 | 50.9 | 40.9 | 46.6 | 71.5 |
| | ✓ | ✓ | ✓ | 82.9 | 80.7 | 67.3 | 76.7 | 77.3 |
| | ✓ | | | 22.9 | 17.2 | 8.9 | 20.8 | 34.4 |
| 57(9.84%) | ✓ | ✓ | | 55.6 | 51.8 | 41.5 | 46.2 | 70.5 |
| | ✓ | ✓ | ✓ | 82.0 | 79.1 | 66.0 | 74.7 | 76.8 |

Table C.1. The impact of the number of masked frames with MTP (Feature-to-Label) on the 50Salads dataset. We report the results of different numbers of masked frames based on the transformer backbone using only the MTP, the MTP combined with Naive, and the CTE after initialization, respectively. On average, each action segmentation contains 579 frames.

ture of flow attention which contains the competition architecture and the allocation architecture.

## C. Ablation studies

**C.1 The impact of the number of masked frames.** To study the effect of the MTP with different masking ratios, we perform the ablation experiments based on the transformer backbone using only the MTP, the MTP combined with the Naive, and the CTE after initialization on the 50Salads dataset, respectively. We first compare the number of masked frames for the MTP with the reconstruction target as the action category (Feature-to-Label). As shown in Tab. C.1, during initializing the segmentation model, increasing the number of masked frames causes a degradation in performance. This drop in performance may be because the masked frames except for annotated frames fail to obtain supervision information of labels, which are not computed as classification loss functions. Compared to masking only annotation frames, combining MTP and Naive produces over-segmentation errors in masking multiple frames, as shown by low F1 scores. After adding CTE, the performance of the model masking multiple frames is significantly improved but is not as good as that of masking only annotated timestamps.

| Number | Initialization | | CTE | F1@{10, 25, 50} | | | Edit | Acc |
|---|---|---|---|---|---|---|---|---|
| | MTP | Naive | | | | | | |
| | ✓ | | | 0.1 | 0.0 | 0.0 | 10.4 | 2.6 |
| 1(0.17%) | ✓ | ✓ | | 53.6 | 49.7 | 39.7 | 44.9 | 71.2 |
| | ✓ | ✓ | ✓ | 83.8 | 82.2 | 69.0 | 76.7 | 77.2 |
| | ✓ | | | 0.4 | 0.2 | 0.0 | 9.7 | 2.5 |
| 3(0.52%) | ✓ | ✓ | | 51.9 | 47.6 | 38.2 | 44.9 | 69.9 |
| | ✓ | ✓ | ✓ | 83.9 | 82.0 | 67.9 | 76.0 | 76.7 |
| | ✓ | | | 0.2 | 0.1 | 0.0 | 9.4 | 2.5 |
| 5(0.86%) | ✓ | ✓ | | 53.4 | 49.2 | 39.1 | 45.1 | 70.1 |
| | ✓ | ✓ | ✓ | 83.4 | 81.4 | 67.8 | 76.6 | 76.5 |
| | ✓ | | | 0.0 | 0.0 | 0.0 | 8.7 | 2.5 |
| 7(1.21%) | ✓ | ✓ | | 53.8 | 49.6 | 39.8 | 45.7 | 70.2 |
| | ✓ | ✓ | ✓ | 83.7 | 81.5 | 67.8 | 76.5 | 76.9 |
| | ✓ | | | 0.0 | 0.0 | 0.0 | 7.7 | 2.4 |
| 13(2.25%) | ✓ | ✓ | | 53.7 | 49.4 | 40.4 | 45.4 | 70.5 |
| | ✓ | ✓ | ✓ | 84.0 | 81.8 | 68.4 | 76.7 | 76.7 |
| | ✓ | | | 0.0 | 0.0 | 0.0 | 6.9 | 2.4 |
| 15(2.60%) | ✓ | ✓ | | 52.0 | 47.6 | 38.5 | 43.1 | 69.8 |
| | ✓ | ✓ | ✓ | 83.8 | 81.6 | 67.6 | 75.8 | 76.9 |
| | ✓ | | | 0.0 | 0.0 | 0.0 | 4.7 | 2.4 |
| 29(5.00%) | ✓ | ✓ | | 51.9 | 47.6 | 38.4 | 43.1 | 69.5 |
| | ✓ | ✓ | ✓ | 82.9 | 80.6 | 67.5 | 76.0 | 76.6 |
| | ✓ | | | 0.0 | 0.0 | 0.0 | 4.7 | 2.4 |
| 57(9.84%) | ✓ | ✓ | | 49.9 | 45.6 | 37.1 | 42.9 | 69.7 |
| | ✓ | ✓ | ✓ | 83.0 | 80.7 | 67.4 | 75.5 | 76.8 |

Table C.2. The impact of the number of masked frames with MTP (Feature-to-Feature) on the 50Salads dataset. We report the results of different numbers of masked frames based on the transformer backbone using only the MTP, the MTP combined with Naive, and the CTE after initialization, respectively. On average, each action segmentation contains 579 frames.

As shown in Tab. C.2, we further evaluate the impact of the number of masked frames for MTP with the reconstruction target as the I3D feature (Feature-to-Feature). Using only the MTP, the initialization model lacks supervision information of action category, which results in few correct frames of the output. By combining the MTP and the Naive, the impact of the number of masked frames for MTP is small on the performance. We also perform experiments based on Feature-to-Feature models with the CTE and they perform approximately as well for different num-

| Method | GTEA | | | | |
|---|---|---|---|---|---|
| | F1@{10, 25, 50} | | | Edit | Acc |
| Naive | 47.4 | 42.1 | 33.0 | 45.1 | 47.6 |
| MTP | 67.6 | 62.7 | 47.3 | 61.0 | 60.5 |
| Naive+MTP | 76.8 | 74.0 | 59.0 | 70.6 | 67.9 |
| MTP+Naive | 77.0 | 72.4 | 56.1 | 70.5 | 65.4 |
| Naive+MTP+CTE | 90.7 | 89.0 | 74.4 | 87.2 | 75.3 |
| MTP+Naive+CTE | 91.5 | 90.1 | 76.2 | 88.5 | 75.7 |

Table C.3. The impact of the order of the MTP and the Naive on the GTEA dataset.

| Method | 50Salads | | | | |
|---|---|---|---|---|---|
| | F1@{10, 25, 50} | | | Edit | Acc |
| Naive | 41.3 | 36.1 | 25.3 | 40.0 | 37.9 |
| MTP | 55.4 | 51.1 | 40.9 | 46.8 | 68.7 |
| Naive+MTP | 62.2 | 58.6 | 48.2 | 53.3 | 75.8 |
| MTP+Naive | 62.6 | 58.1 | 47.4 | 52.6 | 74.0 |
| Naive+MTP+CTE | 84.3 | 82.3 | 72.5 | 77.3 | 80.2 |
| MTP+Naive+CTE | 84.4 | 82.2 | 71.2 | 77.5 | 80.2 |

Table C.4. The impact of the order of the MTP and the Naive on the 50Salads dataset.

| Method | GTEA | 50Salads | Breakfast |
|---|---|---|---|
| Constrained K-medoids [1] | 75.3% | 81.3% | 76.9% |
| MTP+Naive | 82.5% | 87.5% | 82.3% |

Table C.5. Comparison of the accuracy of pseudo-labels between our initialized model and unsupervised method (Constrained K-medoids [1]) on the three datasets.

| $\gamma$ | F1@{10, 25, 50} | | | Edit | Acc |
|---|---|---|---|---|---|
| 0.075 | 83.6 | 81.2 | 67.4 | 77.2 | 77.5 |
| 0.15 | 83.3 | 81.2 | 68.1 | 76.8 | 77.8 |
| 0.5 | 84.0 | 81.6 | 70.3 | 77.7 | 79.0 |
| 0.75 | 83.9 | 81.6 | 70.9 | 77.0 | 79.5 |
| 1 | 84.4 | 82.2 | 71.2 | 77.5 | 80.2 |
| 1.25 | 84.1 | 81.8 | 71.0 | 76.8 | 80.0 |

Table C.6. The impact of $\gamma$ on the 50Salads dataset.

bers of masked frames.

**C.2 The impact of the order of initialization methods.** To study the impact of the order of initialization methods, we evaluate the MTP and the Naive on the GTEA and 50Salads datasets. As shown in Tab. C.3 and Tab. C.4, combining the MTP and the Naive outperforms using only the MTP or the Naive, which demonstrates that they are complementary. Using the Naive before or after the MTP performs similar results. This is because combining the MTP and the Naive to initialize the model can capture contextual information, which helps to identify unlabelled action frames. Furthermore, refining the model by using the CTE achieves better results.

**C.3 Comparison of our initialized model and unsupervised methods.** Without the initializing phase, previous methods train models by generating pseudo-labels through an unsupervised approach [1]. As shown in Tab. C.5, our approach generates better frame-wise pseudo-labels by utilizing the initialized model, which is reflected in the better accuracy of pseudo-labels.

**C.4 The impact of the weight of the segmental confidence loss $\gamma$.** During the refinement phase, we set the weight of the smoothing loss to $\lambda = 0.15$ as in [2], and the weight of the segmental confidence loss $\gamma = 1$. As shown in Tab. C.6, we study the impact of $\gamma$ on the performance of the 50Salads dataset. Reducing $\gamma$ to 0.075 still improves the performance but is not as good as the default value of $\gamma = 1$. Increasing its value to $\gamma = 1.25$ also causes a degradation in performance. This drop in performance may be due to the fact that segmental confidence loss heavily relies on shifted boundaries of pseudo-labels, which affects the recognition of frames near the boundaries.

| | F1@{10, 25, 50} | | | Edit | Acc |
|---|---|---|---|---|---|
| $\mathcal{L}_{wce}$ | 80.6 | 77.0 | 61.3 | 72.7 | 73.9 |
| $\mathcal{L}_{wce}$+$\mathcal{L}_{tmse}$ | 83.0 | 80.1 | 66.3 | 76.9 | 76.1 |
| $\mathcal{L}_{wce}$+$\mathcal{L}_{tmse}$+$\mathcal{L}_{sconf}$ | 84.4 | 82.2 | 71.2 | 77.5 | 80.2 |

Table C.7. The impact of different loss functions on the 50Salads dataset.

| | F1@{10, 25, 50} | | | Edit | Acc |
|---|---|---|---|---|---|
| **GTEA** | | | | | |
| Li et al. [2] | 72.4 | 66.2 | 43.6 | 71.4 | 59.3 |
| EM-TSS [3] | - | - | - | - | - |
| D-TSTAS | 90.1 | 84.3 | 60.5 | 87.2 | 62.9 |
| **50Salads** | | | | | |
| Li et al. [2] | 60.8 | 48.5 | 23.1 | 60.6 | 52.3 |
| EM-TSS [3] | 62.9 | 50.5 | 25.0 | 63.9 | 52.0 |
| D-TSTAS | 77.7 | 62.5 | 31.8 | 74.3 | 57.8 |
| **Breakfast** | | | | | |
| Li et al. [2] | 65.5 | 52.2 | 28.0 | 70.4 | 51.2 |
| EM-TSS [3] | 57.3 | 46.9 | 25.0 | 61.7 | 48.5 |
| D-TSTAS | 67.6 | 54.3 | 31.0 | 69.4 | 52.4 |

Table C.8. Comparison with different methods by using the start frame as the timestamp for each action segment on the three datasets.

| | F1@{10, 25, 50} | | | Edit | Acc |
|---|---|---|---|---|---|
| **GTEA** | | | | | |
| Li et al. [2] | 76.5 | 73.0 | 55.6 | 68.7 | 63.8 |
| EM-TSS [3] | - | - | - | - | - |
| D-TSTAS | 90.1 | 88.8 | 73.3 | 87.0 | 71.5 |
| **50Salads** | | | | | |
| Li et al. [2] | 74.2 | 71.0 | 59.2 | 68.3 | 74.3 |
| EM-TSS [3] | 78.4 | 76.0 | 63.5 | 71.1 | 77.1 |
| D-TSTAS | 84.0 | 81.6 | 70.4 | 77.4 | 79.4 |
| **Breakfast** | | | | | |
| Li et al. [2] | 70.8 | 63.5 | 45.4 | 71.3 | 61.3 |
| EM-TSS [3] | - | - | - | - | - |
| D-TSTAS | 76.4 | 68.5 | 48.1 | 75.1 | 65.7 |

Table C.9. Comparison with different methods by using the center frame as the timestamp for each action segment on the three datasets.

**C.5 The effect of different loss functions.** To refine the model, three loss functions are used, including weighted classification loss ($\mathcal{L}_{wce}$), smoothing loss ($\mathcal{L}_{tmse}$), and segmental confidence loss ($\mathcal{L}_{sconf}$). Tab. C.7 shows the impact of each loss on the 50Salads dataset. While the smoothing loss significantly improves the performance of the model, the addition of the segmental confidence loss still gives an

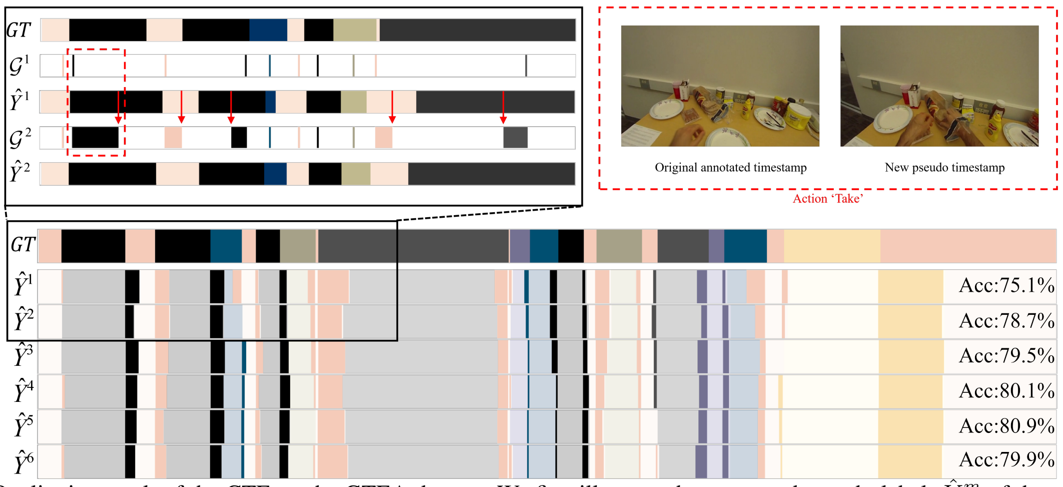

Figure C.1. Qualitative result of the CTE on the GTEA dataset. We first illustrate the generated pseudo-labels $\hat{Y}^m$ of the refining model during different steps $m$ with the CTE. For a clear comparison, we only highlight the wrong action segments. Furthermore, we show the process of updating the progressive pseudo-timestamp groups in step 1. The red arrows denote the new pseudo-timestamps, and we unify the class of frames between the new timestamp and the original timestamp as the category of annotated timestamp. The red box shows the frames of action ('Take') in the pseudo-timestamp group.

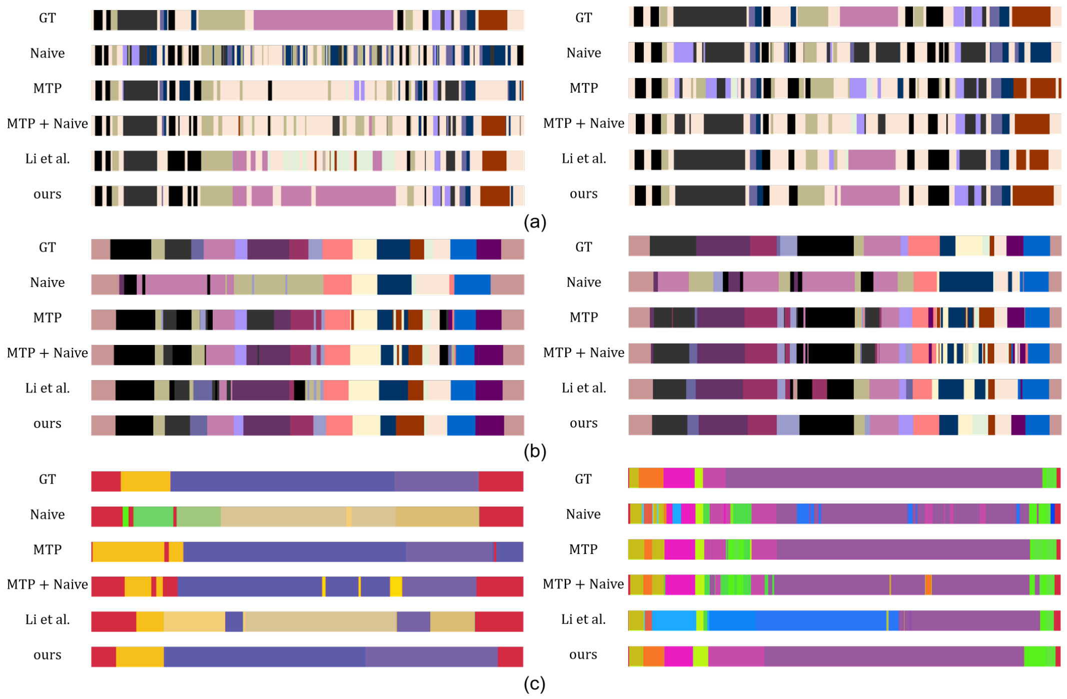

Figure C.2. Qualitative results of Naive, MTP, MTP+Naive, Li et al. [2], and our D-TSTAS on (a) GTEA, (b) 50Salads, and (c) Breakfast datasets.

additional boost in performance. This is because the segmental confidence loss promotes high confidence for all frames within pseudo-timestamp groups and avoids blurring action boundaries.

**C.6 The effect of timestamp locations.** In rare annotation cases, we compare the D-TSTAS with recent approaches [2,3] for timestamp supervision TAS. Specifically, we compare the results of using the start frame and center frame as annotated timestamps for each action segment. Tab. C.8 and Tab. C.9 show our D-TSTAS outperforms the recent methods on the three datasets. While using the center frames of each segment achieves comparable performance to a random frame [2], utilizing the start frames of each segment results in a huge drop in performance. Note that the performance using random annotations is very close to that of real annotations, and the annotating start frames of each segment is a low-quality annotation [2].

## D. Qualitative Analysis

**D.1 Qualitative result of the CTE.** We visualize the generated pseudo-labels of the refining model during 6 iterations with the CTE. As shown in Fig. C.1, our CTE gradually improves the accuracy of the pseudo-labels as the pseudo-timestamp groups are updated. We further show the process of updating steps of the CTE in step 1. These pseudo-timestamp groups obtain rich action representations for action segments, the use of which helps the model to achieve better pseudo-labels. As shown in the red box in Fig. C.1, the frames of the pseudo-timestamp group exhibit different semantics about the action of 'Take'.

**D.2 Qualitative results of D-TSTAS.** As shown in Fig. C.2, we show the qualitative results of different methods on three TAS datasets, including the Naive, the MTP, the MTP+Naive, Li et al. [2] and our D-TSTAS. During the initializing phase, the MTP combined with the Naive reduces focus bias compared to the Naive method. The previous refinement method [2] contains representation bias, which results in identifying the wrong action segments. Our proposed method significantly alleviates both of these biases, and the results of our D-TSTAS are the closest to the ground truth.

## References

[1] Nadine Behrmann, S Alireza Golestaneh, Zico Kolter, Juergen Gall, and Mehdi Noroozi. Unified fully and timestamp supervised temporal action segmentation via sequence to sequence translation. In *European Conference on Computer Vision*, pages 52–68. Springer, 2022. 3

[2] Zhe Li, Yazan Abu Farha, and Jurgen Gall. Temporal action segmentation from timestamp supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8365–8374, 2021. 1, 3, 4, 5

[3] Rahul Rahaman, Dipika Singhania, Alexandre Thiery, and Angela Yao. A generalized and robust framework for timestamp supervision in temporal action segmentation. In *European Conference on Computer Vision*, pages 279–296. Springer, 2022. 3, 4

[4] Haixu Wu, Jialong Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Flowformer: Linearizing transformers with conservation flows. In *International Conference on Machine Learning*, pages 24226–24242. PMLR, 2022. 1, 2

[5] Fangqiu Yi, Hongyu Wen, and Tingting Jiang. Asformer: Transformer for action segmentation. In *The British Machine Vision Conference (BMVC)*, 2021. 1