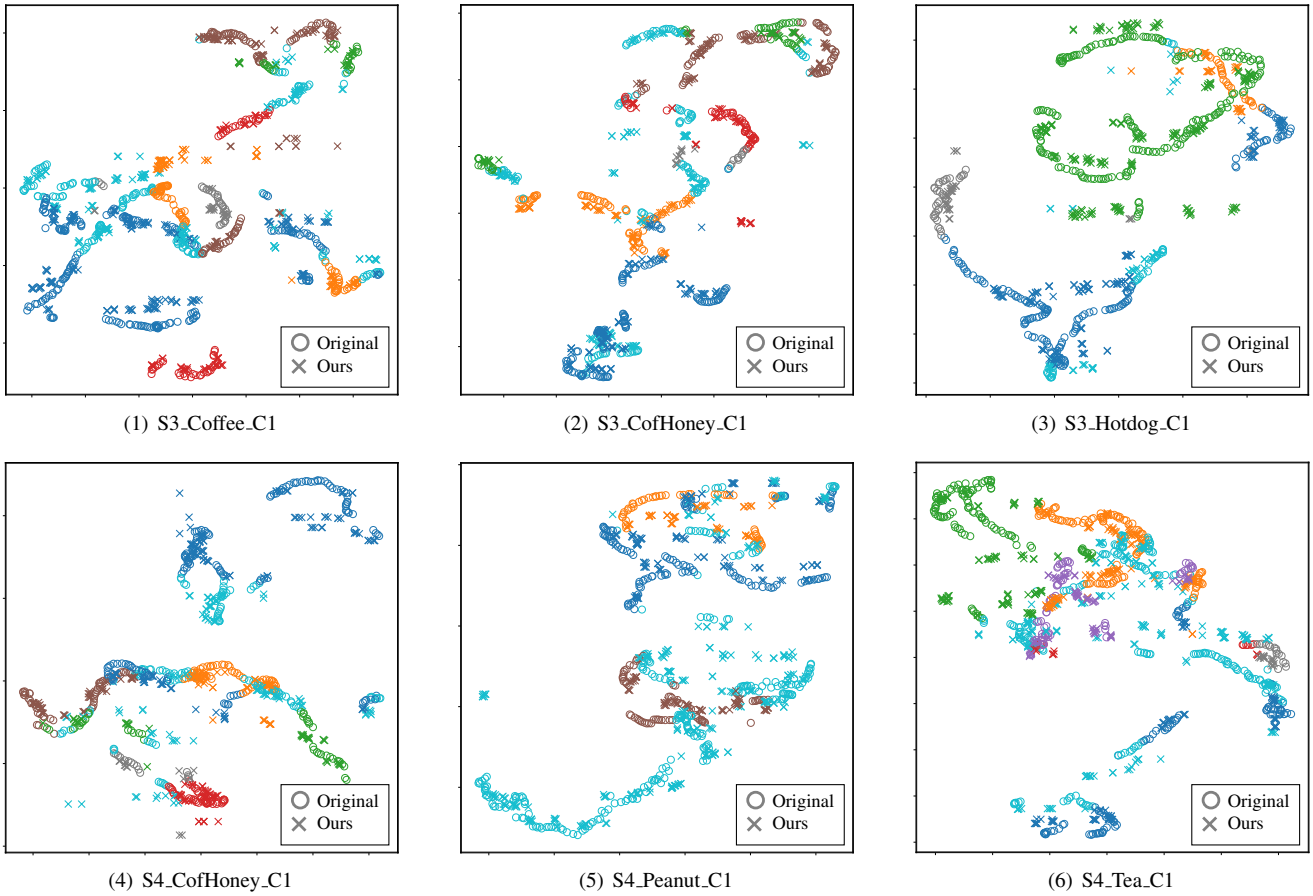
Condensing Action Segmentation Datasets via Generative Network Inversion Supplementary Material

Guodong Ding, Rongyu Chen and Angela Yao
National University of Singapore
{dinggd, rchen, ayao}@comp.nus.edu.sg

	Mean	Coreset	TCA	Encoded	Ours	Encoded†	Ours†
(Condensed) Feature dimension d	2048	2048	256	256	256	256	256
Instance per segment K	1	1	1	8	8	full	full
Sequence sampling ratio γ	1.0	1.0	1.0	0.5	0.5	1.0	1.0

Table A. Detailed experiment settings for the approaches in the main paper.



(1) S3_Coffee_C1 (2) S3_CofHoney_C1 (3) S3_Hotdog_C1

(4) S4_CofHoney_C1 (5) S4_Peanut_C1 (6) S4_Tea_C1

Figure A. More video feature visualizations from the GTEA dataset. Each sub-figure is captioned by the corresponding video name.

A. Experiment Settings

Table A summarizes the detailed configurations of each approach presented in Tab. 1 in the main paper in terms of the condensed feature dimension (d), the number of instances per segment (K) and the sequence sampling ratio (γ). These settings are chosen to vigorously ensure fair comparison between similar storage requirements.

B. TAS Loss Functions

Given a video $\mathbf{X} = \{x^1, ..., x^T\}$ of T frames long, popular TAS works [2–4, 8] formulate segmentation as a classification task to predict the action label for each frame, *i.e.*,

$$y^{1:T} = (y^1, y^2, ..., y^T), \tag{1}$$

where $y^t \in \mathcal{A}$ is the action label for frame at time t. The segmentation model is then trained with a frame-wise cross-entropy loss:

$$\mathcal{L}_{\text{cls}}(x, y) = \frac{1}{T} \sum_t -\log(p^t(y^t)), \tag{2}$$

where $p^t \in \mathbb{R}^A$ is the estimated action probability for the frame x^t. In addition, a smoothing loss is added to encourage smooth transitions between consecutive frames and mitigate the over-segmentation issue:

$$\mathcal{L}_{\text{sm}}(x, y) = \frac{1}{TA} \sum_{t,a} \tilde{\Delta}^2_{t,a}, \quad \tilde{\Delta}_{t,a} = \begin{cases} \Delta_{t,a} : \Delta_{t,a} \leq \tau \\ \tau \quad : \text{otherwise} \end{cases}, \tag{3}$$
$$\Delta_{t,a} = \left| \log p^t(a) - \log p^{t-1}(a) \right|.$$

τ is set to 4 as per [2]. The full training loss is a balanced combination of the above two:

$$\mathcal{L}_{\text{tas}} = \mathcal{L}_{\text{cls}} + \lambda \cdot \mathcal{L}_{\text{sm}}. \tag{4}$$

We set the trade-off parameter $\lambda = 0.15$ following most existing works [2, 8].

C. Visualizations

Fig. A presents more visualizations of the original and generated video features with t-SNE [6] with videos from the GTEA dataset. Most of the generated features are well-aligned with the original.

D. Task Comparison

Table B highlights the key differences between two video understanding tasks: action recognition (AR) and temporal action segmentation (TAS). Several notable distinctions emerge. **First**, AR models typically operate on raw video frames, such as RGB frames or derived optical flow, whereas TAS models often utilize pre-computed frame features. This choice facilitates easier access and ensures fair comparisons across different methods [1]. **Second**, AR models usually handle a fixed number of input frames. For instance, C3D [5] splits videos into clips with a predefined frame count, *i.e.*, $N = 16$. In contrast, TAS models process entire videos of varying lengths at their full temporal resolution, without subsampling. **Lastly**, the output dimensionality of these tasks also differs. AR models predict a single, global label for the entire set of input frames, which is independent of the number of frames. On the other hand, TAS models preserve the temporal resolution of the input and produce frame-wise predictions, ensuring alignment between input and output dimensions. These distinctions make the adaptation of existing work [7] to TAS non-trivial and call for task-specific designs for the TAS dataset condensation problem.

Task	Input Data	Input Dim	Output Dim
AR	RGB image	$\mathbb{R}^{N\times 3\times H\times W}$	$\mathbb{R}^{1\times C}$
TAS	I3D feature	$\mathbb{R}^{T\times D}$	$\mathbb{R}^{T\times C}$

Table B. Task Comparison between action recognition (AR) and temporal action segmentation (TAS). N is the number of selected input frames. H, W are the height and weight of the frame. T, D are the temporal length and feature dimension of pre-computed video features, respectively. C is the number of action classes.

References

[1] Guodong Ding, Fadime Sener, and Angela Yao. Temporal action segmentation: An analysis of modern techniques. *IEEE TPAMI*, 2023. 2

[2] Yazan Abu Farha and Jurgen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *CVPR*, 2019. 2

[3] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *CVPR*, 2017.

[4] Dipika Singhania, Rahul Rahaman, and Angela Yao. C2f-tcn: A framework for semi-and fully-supervised temporal action segmentation. *IEEE TPAMI*, 2023. 2

[5] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 2

[6] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 2

[7] Ziyu Wang, Yue Xu, Cewu Lu, and Yong-Lu Li. Dancing with still images: Video distillation via static-dynamic disentanglement. In *CVPR*, 2024. 2

[8] Fangqiu Yi, Hongyu Wen, and Tingting Jiang. Asformer: Transformer for action segmentation. In *BMVC*, 2021. 2