A. Additional Experiments

A.1. Actionformer Downstream Head

Additional results using Actionformer [11], a recently introduced state-of-the-art TAL head, is presented in Table 3. Results using GTAD and BMN are also shown in parallel with the Actionformer results in Table 3 as exactly the same feature is utilized. Consistent performance improvement is observed with the adoption of SoLa strategy regardless of the downstream heads.

A.2. Additional Ablation Study

Inspired by Adapters [7, 8], the SoLa module has two parallel passes: a skip connection and a 1D CNN pass with a learnable scalar gating parameter initialized as 0. This structure makes the SoLa module the identity mapping at its first stage of training. As the training precedes, the gating parameter will deviate from 0 and the SoLa module will start to slowly enhance the input feature sequence. Table 1 demonstrates the ablation study of the design choice. We can see that while the SoLa module with a direct pass works reasonably well, the Adaptor style SoLa module brings additional performance gain thanks to its conservative enhancement of the feature sequence.

Additional ablation studies on the remaining design choices of the SoLa strategy are presented in Table 5 and 6. It is worth noting that, results on varying K values suggest that appropriate Λ assignment is an essential part in terms of achieving *temporally sensitive* snippet feature sequences, which justifies our λ function design. Moreover, result in Table 6 alludes that asymmetric projector significantly stabilizes the training procedure.

A.3. Error bars of the main results

To validate the robustness of our method, we report the main results' error bars with standard deviations in Figure 1. 12 and 5 independent downstream head training with varying random seeds was done in Activitynet1.3 [1] and HACS [12] experiments respectively. The result shows that our method significantly outperforms the baselines with a strong statistical significance.

B. Connection to the Contrastive Learning

In this section, we provide a connection between the well-known contrastive learning loss function and our Similarity Matching loss.

Due to the *unlabeled* target dataset assumption, the training of the SoLa module must be done in a self-supervised manner. Since recent studies [4, 5] have shown remarkable success of contrastive learning in the self-supervised representation learning domain, it is natural to start with the standard NT_XENT loss [2]. For the positive sample representation pair (z_i, z_j) and the similarity measure $\mathrm{sim}(\cdot,\cdot)$,

Method	Temporal Action Localization (GTAD)				
	mAP@0.5	@0.75	@0.95	Avg	gain
Baseline	49.78	34.46	7.96	33.84	-
SoLa (Direct)	50.74	35.35	8.29	34.66	+0.78
SoLa (Adapter)	51.17	35.70	8.31	34.99	+1.15

Table 1. Downstream head performance (G-TAD) with respect to the design choices.

Loss	Temporal Action Localization (GTAD)				
	mAP@0.5	@0.75	@0.95	Avg	gain
Baseline	49.78	34.46	7.96	33.84	-
$\mathcal{L}_{in}^{\text{soften}}$	45.99	30.60	7.17	30.34	-3.50
$\mathcal{L}_{out}^{\text{soften}}$	50.21	35.14	8.40	34.40	+0.56
$\mathcal{L}^{\text{SM}}$	51.17	35.70	8.31	34.99	+1.15

Table 2. Downstream head performance (G-TAD) with respect to the loss terms. For $\mathcal{L}_{out}^{\text{soften}}$ and $\mathcal{L}_{in}^{\text{soften}}$, they share the λ assignment setting with the main experiments.

$\mathcal{L}_{i,j}^{\text{NT\_XENT}}$ is defined as follows:[1]

$$\mathcal{L}^{\text{NT\_XENT}}(z_i, z_j) = \underbrace{-\,\mathrm{sim}(z_i, z_j)}_{L_{alignment}} \underbrace{+\log\Big(\exp(\mathrm{sim}(z_i, z_j)) + \sum_{k\in I\setminus\{i,j\}} \exp(\mathrm{sim}(z_i, z_k))\Big)}_{L_{distribution}}, \tag{1}$$

where I is the index set containing all the sample index. However, unlike the common contrastive learning setting, we cannot exploit data augmentation to generate positive samples from the given data since we do not have any means to manipulate the feature-level sample without hurting its essence. This makes the standard positive/negative pair concept inapplicable and thereby requires devising a different approach for training the SoLa module.

In this regard, we start with the concept of the temporal structure that videos naturally convey: "adjacent frames should be similar, while remote frames should remain distinct". In fact, the instantiation of the above temporal structure can substitute the standard positive/negative pair concept. But unlike the discrete positive/negative distinction in Equation (1), the temporal structure offers a *softened* version of the distinction in that "the similarity decreases as the distance between the two features increases". Therefore, we introduce a softened indicator function $\lambda(\cdot,\cdot) \to [0,1]$, whose output represents the input pair's *positiveness* by a continuous real value. For instance, a high $\lambda(z_i, z_j)$ value

[1] $\mathcal{L}^{\text{NT\_XENT}}$ expansion is from [3, 9]

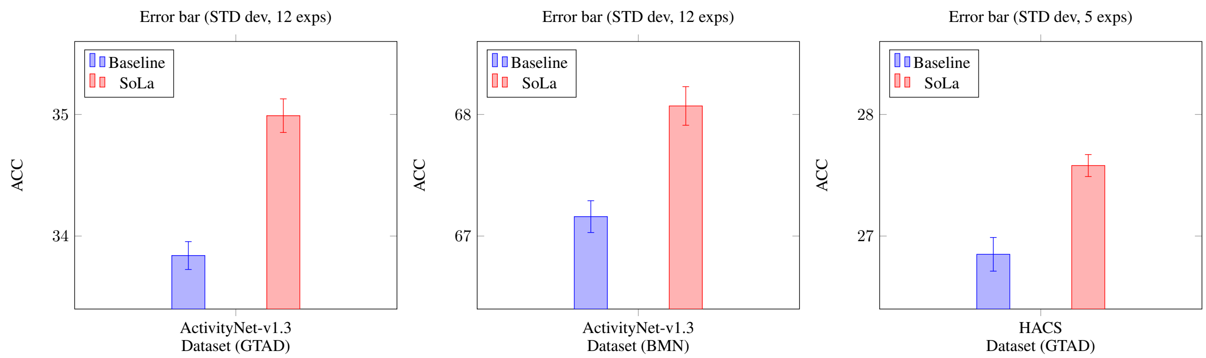

Figure 1. Error bars of main experiments.

indicates (z_i, z_j) to be treated more like a semantically close pair, whereas a low $\lambda(z_i, z_j)$ value represents larger semantic distance.

To incorporate $\lambda(\cdot,\cdot)$ into Equation (1), we pay careful attention to the following two points: (i) $L_{alignment}$ and $L_{distribution}$ should be more influential with high $\lambda(z_i, z_j)$ and low $\lambda(z_i, z_k)$ respectively to match our soft λ positiveness concept, and (ii) Equation (1) must be recovered from Equation (2) and (3) by setting $\lambda(\cdot,\cdot)$ as a discrete indicator function that only returns 1 if the given pair is a positive pair and 0 otherwise. From these points, we placed the coefficients $\lambda(z_i, z_j)$ and $1-\lambda(z_i, z_k)$ in front of the positive/negative pairs in Equation (1). Two possible softened NT_XENT losses can be derived from the coefficients' positions:

$$\mathcal{L}_{in}^{\text{soften}}(z_i, z_j) = \underbrace{-\lambda(z_i, z_j)\,\text{sim}(z_i, z_j)}_{L_{alignment}} + \underbrace{\log\Big(\exp(\lambda(z_i, z_j)\text{sim}(z_i, z_j)) + \sum_{k\in I\setminus\{i,j\}} \exp((1-\lambda(z_i, z_k))\text{sim}(z_i, z_k))\Big)}_{L_{distribution}}, \tag{2}$$

$$\mathcal{L}_{out}^{\text{soften}}(z_i, z_j) = \underbrace{-\lambda(z_i, z_j)\,\text{sim}(z_i, z_j)}_{L_{alignment}} + \underbrace{\log\Big(\lambda(z_i, z_j)\exp(\text{sim}(z_i, z_j)) + \sum_{k\in I\setminus\{i,j\}} (1-\lambda(z_i, z_k))\exp(\text{sim}(z_i, z_k))\Big)}_{L_{distribution}}. \tag{3}$$

We observed that $\mathcal{L}_{out}^{\text{soften}}$ performs well, whereas $\mathcal{L}_{in}^{\text{soften}}$ degrades the downstream task performance significantly (Table 2). Note that the gradient with respect to $\text{sim}(z_i, z_k)$ in $\mathcal{L}_{in}^{\text{soften}}$'s $L_{distribution}$ term is scaled by $1-\lambda$. As $1-\lambda \in [0,1]$, we attribute the deterioration of the performance to the broken gradient balance between $L_{alignment}$ and $L_{distribution}$ terms in $\mathcal{L}_{in}^{\text{soften}}$.

While we can directly work with the Equation (3), another interesting observation in Equation (3) is that it results in a non-zero term even if only one pair (z_i, z_j) is given for its computation, while a trivial cancellation of $L_{alignment}$ and $L_{distribution}$ occurs in Equation (1) and (2) . As $I\setminus\{i,j\} = \emptyset$ in the one pair case, the single pair $\mathcal{L}_{out}^{\text{soften}}$ can be represented as follows:

$$\begin{aligned}
&\mathcal{L}_{out}^{\text{soften}}(z_i, z_j) \\
&= -\lambda(z_i, z_j)\text{sim}(z_i, z_j) + \log\Big(\lambda(z_i, z_j)\exp(\text{sim}(z_i, z_j))\Big) \\
&= -\lambda(z_i, z_j)\text{sim}(z_i, z_j) + \log\lambda(z_i, z_j) + \text{sim}(z_i, z_j) \\
&= \big(1-\lambda(z_i, z_j)\big)\text{sim}(z_i, z_j) + const \\
&= -(1-\lambda(z_i, z_j))\log(1-p) + const,
\end{aligned} \tag{4}$$

where $p = 1-\exp(-\text{sim}(z_i, z_j))$. Note that p monotonically increases as $\text{sim}(z_i, z_j)$ increases for all $\text{sim}(z_i, z_j)$, allowing the interpretation of p as $\text{sim}(z_i, z_j)$ without the loss of generality. However, the single pair $\mathcal{L}_{out}^{\text{soften}}$ goes to 0 when the λ goes to 1, regardless of the p value. To resolve this issue, we added symmetric term $-\lambda(z_i, z_j)\log p$ to the single pair $\mathcal{L}_{out}^{\text{soften}}$. Here, if we assume the strictly positive similarity measure (e.g., $\text{sim}(z_1, z_2) = \frac{1}{\|z_1 - z_2\|^2}, z_1 \neq z_2$), p is bounded to $(0,1)$. Denoting the bounded p as $\hat{p}$, similarity matching loss $\mathcal{L}^{\text{SM}}$ is formulated as

$$\mathcal{L}^{\text{SM}}(z_i, z_j) = -\lambda(z_i, z_j)\log\hat{p} - (1-\lambda(z_i, z_j))\log(1-\hat{p}), \tag{5}$$

where $\hat{p} \in (0,1)$ is the network's prediction of the given pair's similarity. Intuitively, our loss term simply minimizes the Binary Cross Entropy (BCE) between the network prediction $\hat{p}$ and the given label $\lambda(z_i, z_j)$, as its name "similarity matching" suggests. We empirically found out that our sim-

Method	Temporal Action Localization (GTAD)					Temporal Action Localization (ActionFormer)					Temporal Action Proposal (BMN)				
	mAP@0.5	@0.75	@0.95	Avg	gain	mAP@0.5	@0.75	@0.95	Avg	gain	AR@1	@10	@100	AUG	gain
Baseline	49.78	34.46	7.96	33.84	-	50.22	33.81	7.75	33.19	-	33.59	56.79	75.05	67.16	-
SoLa(ours)	51.17	35.70	8.31	34.99	+1.15	51.64	34.81	8.02	34.21	+1.02	34.25	57.75	75.86	68.07	+0.91

Table 3. TAL performance in Activitynet1.3 [1] dataset with various downstream heads.

Hyperparameter	Value
Epoch	500
Learning rate	0.0001
Hidden Units	1024
Conv layers	3
Kernel sizes	{5, 1, 1}
Optimier	AdamW [6]
Batch size	256
K	16
s	8
TSM size	8×8

(a) hyperparameters for the SoLa training.

Hyperparameter	Value
Learning rate	0.01
Momentum	0.9
Epoch	100
Batch size	256
Optimizer	SGD

(b) Hyperparameters for the linear evaluation

Table 4. SoLa module hyperparameters. K is a constant for the λ assignment (Equation 1 in the main paper) and s is the step size described in the caption of the main paper's Figure 2.

Setting		Temporal Action Localization (GTAD [10])			
K	Step	mAP@0.5	@0.75	@0.95	Avg
Baseline		49.78	34.46	7.96	33.84
4	8	50.16	34.92	8.17	34.35
8	8	50.59	35.29	7.94	34.53
32	8	50.80	35.33	8.56	34.80
16	2	50.63	35.39	8.23	34.69
16	4	50.96	35.61	7.65	34.83
16	12	50.88	35.37	7.58	34.64
16	8	51.17	35.70	8.31	34.99

Table 5. Additional ablation study results on ActivityNet-v1.3.

Architecture	Temporal Action Localization (GTAD)			
	mAP@0.5	@0.75	@0.95	Avg
Symmetric	49.75	34.58	6.90	33.73
Asymmetric	51.17	35.70	8.31	34.99

Table 6. Ablation study results about asymmetric projector on ActivityNet1.3.

ilarity matching loss even works better when it comes to optimizing the soft landing module.

C. Justification of the similarity assumption

Assignment of soft pseudo-label (Eq.1) is from the local similarity assumption:

> *"Only adjacent features should be similar while distant features remain distinct.",*

which is based on the empirical observation on general videos. One might suspect that the above assumption oversimplifies the complex characteristics of untrimmed videos. However, the fact that only "sampled subsequence of the given video" is utilized in the SoLa training procedure should not be neglected. It indicates that for the SoLa training, the assumption does not have to hold in the whole video, but only in the sampled video clip which is relatively shorter than the whole video. Thus, the exact local similarity assumption utilized in the SoLa training procedure should be noted as

> *"Only adjacent features should be similar while distant features remain distinct, if both features are from the same sampled subsequence.",*

which is a far more relaxed one compared to that of its whole-video version. It's worth noting that adjusting the length of the sampled subsequence can accommodate videos with repetitions because if the subsequence length is much shorter than the repetition duration, the local similarity assumption still holds.

Moreover, there is no need for all the samples to strictly satisfy the local similarity assumption. Although there might be some counterexamples, we have found that training with the above assumption is reasonable as long as the number of them does not exceed the samples that obey the local similarity assumption. To support our claim, empirical analysis on this issue is presented in our main paper (Figure 3) which shows that general and widely used untrimmed video datasets mostly follow the local similarity assumption.

D. Implementation Details

We adopted a simple feedforward neural network for the SoLa module architecture. It consists of three layers: 1DConv-ReLU-1DConv-ReLU-1DConv, with additional residual connection from the very first layer (before the first 1DConv) and the last layer (after the last 1DConv). Here, the 1DConv pass is scaled with α, a learnable parameter which is initialized as 0. Other detailed hyperparameters are presented in Table 4.

We do not think our SoLa module's architecture is globally optimal. Rather, we show that despite its overly simplified architecture, the SoLa strategy still works well. Future research will include devising better SoLa module architecture.

References

[1] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015. 1, 3

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 2020. 1

[3] Ting Chen, Calvin Luo, and Lala Li. Intriguing properties of contrastive losses. *Advances in Neural Information Processing Systems*, 2021. 1

[4] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 1

[5] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 2020. 1

[6] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 3

[7] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Learning multiple visual domains with residual adapters. *Advances in neural information processing systems*, 30, 2017. 1

[8] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Efficient parametrization of multi-domain deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8119–8127, 2018. 1

[9] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, 2020. 1

[10] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. G-tad: Sub-graph localization for temporal action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 3

[11] Chenlin Zhang, Jianxin Wu, and Yin Li. Actionformer: Localizing moments of actions with transformers. In *European Conference on Computer Vision*, 2022. 1

[12] Hang Zhao, Antonio Torralba, Lorenzo Torresani, and Zhicheng Yan. Hacs: Human action clips and segments dataset for recognition and temporal localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019. 1