Supplementary: Temporal Action Detection with Proposal Denoising Diffusion

Sauradip Nag[1,2] * Xiatian Zhu[1,3] Jiankang Deng[4] Yi-Zhe Song[1,2] Tao Xiang[1,2]

[1] CVSSP, University of Surrey, UK [2] iFlyTek-Surrey Joint Research Center on Artificial Intelligence, UK

[3] Surrey Institute for People-Centred Artificial Intelligence, UK [4] Imperial College London, UK

1. More Implementation Details

1.1. Formulation of Diffusion Model

We provide a detailed review of the formulation of diffusion models, following the notion of [7, 4, 11]. Starting from a data distribution $\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0)$, we define a forward Markovian noising process q which produces data samples $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, ..., $\boldsymbol{x}_T$ by gradually adding Gaussian noise at each timestep t. In particular, the added noise is scheduled by the variance $\beta_t \in (0,1)$:

$$q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) := \prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \tag{1}$$

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) := \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \boldsymbol{I}) \tag{2}$$

As noted by Ho *et al.* [7], we can directly sample data $\boldsymbol{x}_t$ at an arbitrary timestep t without the need of applying q repeatedly:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) := \mathcal{N}(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\boldsymbol{I}) \tag{3}$$

$$:= \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \epsilon\sqrt{1-\bar{\alpha}_t}, \epsilon \in \mathcal{N}(0, \boldsymbol{I}) \tag{4}$$

where $\bar{\alpha}_t := \prod_{s=0}^{t} \alpha_s$ and $\alpha_t := 1-\beta_t$. Then, we could use $\bar{\alpha}_t$ instead of β_t to define the noise schedule.

Based on Bayes' theorem, it is found that the posterior $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$ is a Gaussian distribution as well:

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_{t-1}; \tilde{\mu}(\boldsymbol{x}_t, \boldsymbol{z}_0), \tilde{\beta}_t \mathbf{I}) \tag{5}$$

where

$$\tilde{\mu}_t(\boldsymbol{x}_t, \boldsymbol{x}_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\boldsymbol{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{x}_t \tag{6}$$

and

$$\tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \tag{7}$$

are mean and variance of this Gaussian distribution.

We could get a sample from $q(\boldsymbol{x}_0)$ by first sampling from $q(\boldsymbol{x}_T)$ and running the reversing steps $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ until $\boldsymbol{x}_0$. Besides, the distribution of $q(\boldsymbol{x}_T)$ is nearly an isotropic Gaussian distribution with a sufficiently large T and reasonable schedule of β_t ($\beta_t \to 0$), making it trivial to sample $\boldsymbol{x}_T \sim \mathcal{N}(0, \boldsymbol{I})$. Moreover, since calculating $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ exactly should depend on the entire data distribution, we could approximate $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ using a neural network, which is optimized to predict a mean μ_θ and a diagonal covariance matrix Σ_θ:

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) := \mathcal{N}(\boldsymbol{x}_{t-1}; \mu_\theta(\boldsymbol{x}_t, t), \Sigma_\theta(\boldsymbol{x}_t, t)) \tag{8}$$

Instead of directly parameterizing $\mu_\theta(\boldsymbol{x}_t, t)$, Ho *et al.* [7] found that learning a network $f_\theta(\boldsymbol{x}_t, t)$ to predict the ϵ or $\boldsymbol{x}_0$ from 4 works the best. We hence choose predicting $\boldsymbol{x}_0$ in this work.

1.2. Detection Decoder

To help understand our detection decoder better, we provide more details below. There are two attention modules in the decoder: (1) The proposed selective attention module and (2) a cross-attention module. As shown in Fig. 1, first we project the noisy queries as noisy action segments both from current time step x_t and previous denoised step x_{t+1} (denoted as reference segment). We calculate the most promising noisy query to be denoised using Eq. (6) and Eq. (7) (in main paper). Additionally, the noisy query at x_t also predicts soft attention weights by a fully-connected layer. With the soft-attention, the decoder samples the most confident query features $\hat{q}_i$ at each decoder layer for cross-attention. Another input of the cross-attention in the decoder is the video encoder features (*i.e.* the condition). Finally the cross-attended noisy query features are passed onto the FFN layers for estimating the noise for denoising.

1.3. Selective Conditioning Mechanism

Our DiffTAD denoiser (refer to Fig 1) houses a transformer-decoder which solves the purpose of both denoising and cascading proposal refinement. The later is done using a novel *selective conditioning* mechanism. Let us restate how selective conditioning works: **(a)** Given an

[*] This work was done during internship with Jiankang Deng.

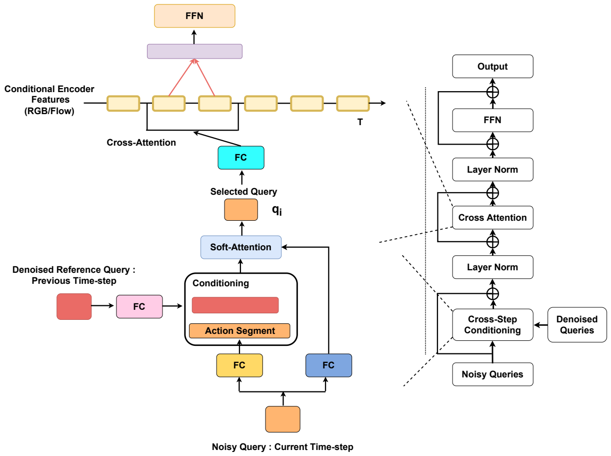

Figure 1. Illustration of detection decoder module in DiffTAD.

action proposal $\psi_t \in \mathbb{R}^{N\times 2}$ at timestep t and $\psi_{t+1} \in \mathbb{R}^{N\times 2}$ the denoised proposals from previous time step, we project them into continuous embedding using Eq (4), denoted as x_t/x_{t+1}. **(b)** We further use a 1-D convolution to project x_t/x_{t+1} into query segments (as shown in Fig 1 in supplementary). We then construct a similarity matrix A using the cosine-similarity between the segments and use Eq (6) to obtain the query set $\hat{P}_{sim}$ having high similarity across timesteps. **(c)** We then construct an IOU-based matrix B, estimated by calculating the IOU overlap of the embedding segments across the time steps using Eq (7), to obtain a query-set denoted by $\hat{P}_{iou}$. **(d)** Finally, we obtain the selective query-set by $\hat{P}_{sim}/\hat{P}_{iou}$ which is then used to vote the most-probable query for denoising.

1.4. Details on the Baseline Model

Since DiffTAD is the first generative TAD model, we adapted DiffusionDet [3] as a baseline generative model for fair comparison. Concretely, given a video, we first process it with a I3D network and flatten to form 1-D features. We then exert a feature pyramid network with several temporal convolutions. Once noise is injected to the action proposals, they are then used to crop the backbone features. Finally, using a dynamic localization and action classification head, we obtain the final output detection.

1.5. Label Assignment in DiffTAD

Similar to RTD-Net [13] the ground-truth instance set $\hat{\psi} = \{\hat{\psi}_n = (\hat{t}^n_s, \hat{t}^n_e, \hat{y}^n)\}^{N_g}_{n=1}$ is composed of N_g targets, where $\hat{t}^n_s$ and $\hat{t}^n_e$ are the starting and ending temporal locations of $\hat{\psi}_n$ and $\hat{y}_n$ is the action label of the corresponding proposal. Likewise, the prediction set of N_p samples is denoted as $\Psi = \{\hat{\psi}_n = (t^n_s, t^n_e, y^n)\}^{N_p}_{n=1}$. We assume N_p is larger than N_g and augment $\hat{\psi}$ to the size of N_p by padding ϕ.

1.6. Details on the TAD Loss

Similar to DETR based TAD designs [13, 12, 10], we have three heads for each noisy query and the set-prediction [8] loss terms for each of them. The set-prediction loss requires pairwise matching cost between the predictions and ground truth instances, taking into account both the category and proposal predictions. The matching cost is formulated as:

$$\mathtt{C} = \lambda_{cls}C_{cls} + \lambda_{l1}C_{l1} + \lambda_{tiou}C_{tiou} + \lambda_{comp}C_{comp} \quad (9)$$

where C_{cls} is the focal loss [9] between the prediction and ground truth class labels. Besides, our action proposal loss contains the common l1-loss C_{l1} and temporal IoU (tIoU) loss C_{tiou} [13]. We additionally have an action completeness loss C_{comp} – a variant of tIOU loss [13] that refines the low confidence proposals. $\lambda_{cls}, \lambda_{l1}, \lambda_{tiou}$ and $\lambda_{comp} \in \mathbb{R}$ are the weights of each component for balancing the multiple losses. Following [13], we adopt $\lambda_{cls} = 1.0$, $\lambda_{l1} = 1.0$, $\lambda_{iou} = 1.0$ and $\lambda_{comp} = 1.0$. We assign multiple predictions to each ground truth with the optimal transport approach [5, 6]. Specifically, for each ground-truth, we select the top-k predictions with the least matching cost as its positive samples, and the others as negatives. Overall, DiffTAD is optimized with a multi-task loss function:

$$\mathtt{L} = \lambda_{cls}L_{cls} + \lambda_{l1}L_{l1} + \lambda_{iou}L_{iou} + \lambda_{comp}L_{comp} \quad (10)$$

The component of training loss is the same as the matching cost, except that the loss is only performed on the matched pairs.

2. More Model Ablation

Importance of positional encoding In this section, we show the importance of temporal positional embedding in the video encoder module. We experiment with removing positional embedding at MLP encoder or directly adding it into the encoder. We contend that the commonly used positional encoding, however, does not bring performance gain. We postulate that the projection using convolutions as well as the depthwise convolutions in the Transformer encoder blocks already leak the location information, as reported in [16]. The results in Table 1 show that the model performance even decreases by 1.3% on avg mAP from using temporal positional embedding in the encoder.

Table 1. Positional encoding (PE) with DiffTAD on THUMOS14.

Model	0.3	0.5	0.7	Avg
Ours w/ PE	72.7	70.1	56.2	66.7
Ours w/o PE	**74.9**	**71.2**	**58.5**	**68.0**

Query padding strategy As introduced in Section 3.3, we need to pad additional proposals to the original ground truth

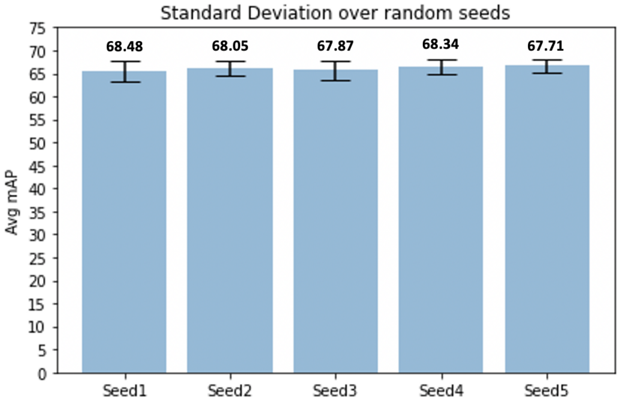

Figure 2. **Statistical results** over 5 independent training instances, with each evaluated 10 times with different random seeds on THUMOS14 dataset. The numbers inside the figure are the mean values.

proposals so that each video has the same number of proposal queries during training N_{train} and evaluation N_{eval}. We study different padding strategies, including (1) repeating original ground truth proposals projected as queries evenly until the total number reaches pre-defined value N_{train}; (2) Padding random queries in Gaussian distribution; (3) Padding random queries following uniform distribution. As shown in Table 2, concatenating uniform random query works the best for DiffTAD, which is different to object detection [3] with Gaussian distributed padding the best. We use the uniform padding strategy as default.

Table 2. Strategies of padding queries on THUMOS14.

Type	0.3	0.5	0.7	Avg
Repeat	70.2	67.8	54.1	63.7
Uniform	**74.9**	**71.2**	**58.5**	**68.0**
Gaussian	74.0	70.5	57.6	67.1

Random seed DiffTAD is given random action proposals as input at the start of inference. One may ask whether there is a large performance variance across different random seeds. We evaluate the stability of DiffTAD by training five models independently with strictly the same configurations except for random seed on THUMOS14 dataset. Then, we evaluate each model instance with ten different random seeds to measure the distribution of performance, inspired by [3]. As shown in Figure 2, most evaluation results are distributed closely to 68.11 avg mAP. Besides, the mean values are all above 67.7 avg mAP, with marginal performance differences across different model instances. This demonstrates that DiffTAD is robust to the random proposals, able to produce reliable performance.

Effect of video feature We evaluate the effect of video features in DiffTAD. For this experiment, we considered two video feature backbones namely R(2+1)D [15] and I3D [2]. It is observed from Table 3 that in both cases DiffTAD outperforms existing approaches consistently on THUMOS dataset. This indicates that the advantage of our model is video feature generic.

Table 3. Effect of video features with DiffTAD on THUMOS14.

Features	**mAP**			
	0.3	**0.5**	**0.7**	**Avg**
I3D [2]	74.9	71.2	58.5	68.0
R(2+1)D [15]	67.4	62.8	48.5	55.9

Ablation on feature-fusion strategy For more extensive evaluation, we further tested the `middle fusion` strategy where both RGB and Flow features share the same denoiser weights. Table 4 shows that the middle and late fusion strategies perform similarly with the former having less parameters. A plausible explanation is that with conditional diffusion modeling, fusing RGB and flow features (*i.e.*, early fusion) makes an over-complex condition for the model to leverage, leading to a more challenging optimization problem.

Table 4. Ablation on fusion strategies on THUMOS dataset

Fusion Strategy	mAP @0.5	Avg mAP
Early	71.8	66.0
Middle	**75.8**	67.9
Late	74.9	**68.0**

3. Applications

Our design can be easily extended for solving applications in a diverse range of fields, including object detection, sound event detection (SED), and 3D action detection (3D-TAD). In the realm of object detection, it can be extended by simply changing the video encoder and TAD decoder with image encoder and image decoder. Instead of denoising 1-D object proposals , it denoises 2-D object bounding boxes. In similar spirit, it can also be extended to the 3D variant of TAD where the task is to localize the temporal actions of 3D human motions. Since diffusion models [14] excel at capturing intricate spatio-temporal dynamics inherent in human actions, this is a possible extension of our work. Finally, as illustrated in the work of DiffSED[1], our DiffTAD design can also be easily extended to detect sound event activity from raw audio using denoising diffusion thus proving the generality of our proposed model design.

4. Visualization

We visualize our sampling step of DiffTAD in Figure 3. The model runs with N proposals. For better visualization, we only draw 30 proposals in the video.
(a) Initial random proposals sampled from the Gaussian distribution are input into the detection detector.

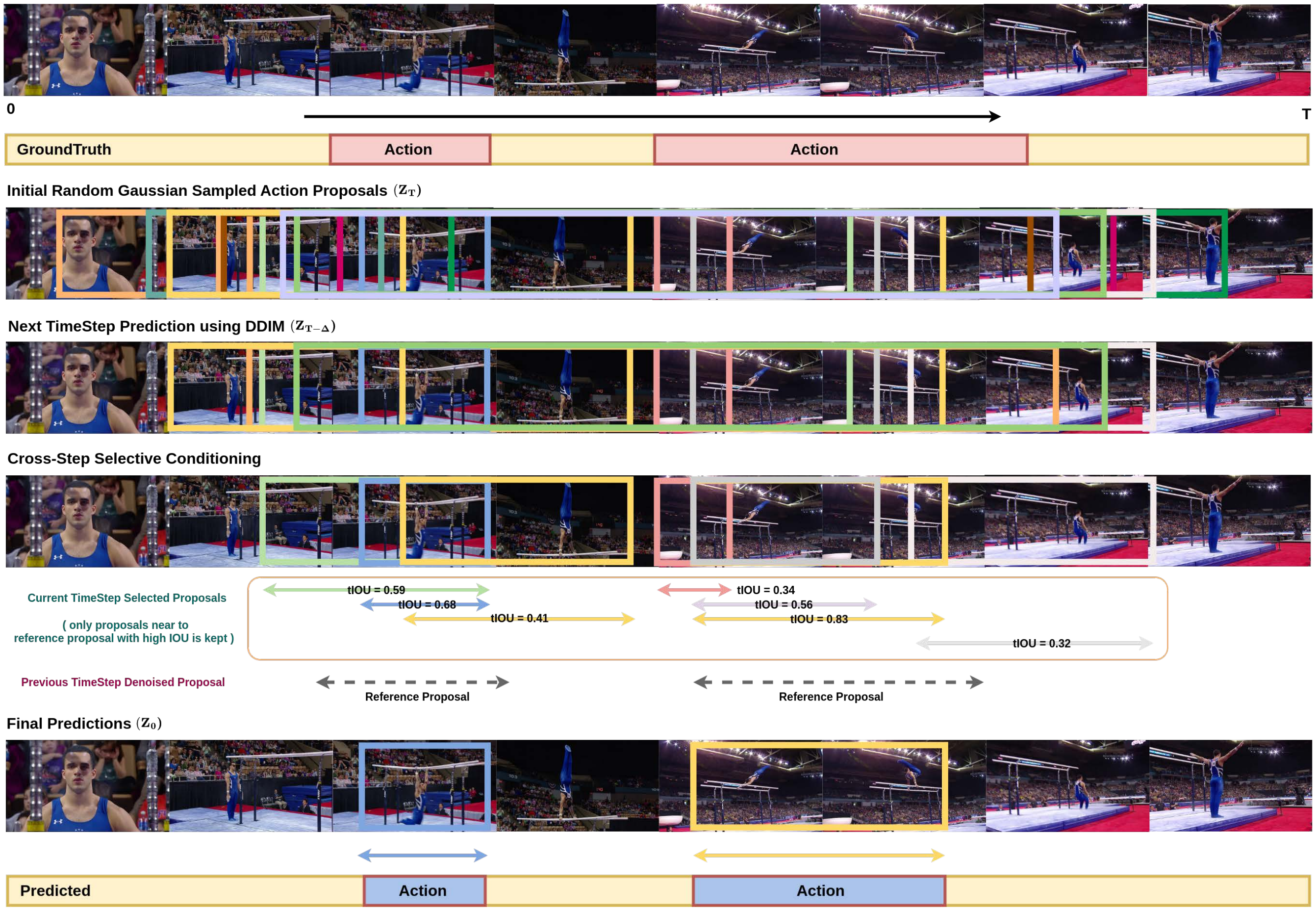
0
T
GroundTruth
Action
Action
Initial Random Gaussian Sampled Action Proposals $(\mathrm{Z_T})$
Next TimeStep Prediction using DDIM $(\mathrm{Z_{T-\Delta}})$
Cross-Step Selective Conditioning
Current TimeStep Selected Proposals
(only proposals near to reference proposal with high IOU is kept)
tIOU = 0.59
tIOU = 0.68
tIOU = 0.41
tIOU = 0.34
tIOU = 0.56
tIOU = 0.83
tIOU = 0.32
Previous TimeStep Denoised Proposal
Reference Proposal
Reference Proposal
Final Predictions $(\mathrm{Z_0})$
Predicted
Action
Action

Figure 3. **Visualization of DiffTAD** proposal denoising step during inference

(b) The detection decoder predicts the category scores and proposal coordinates of the current step. In sub-figure (b), the color brightness is proportional to the score value, where the deep red is high score, and white is low score.
(c) DDIM estimates the proposals for the next step.
(d) Those proposals with lower scores than the threshold are dropped.
(e) New random proposals sampled from the Gaussian distribution are concatenated to the remaining proposals. This new set of proposals are input to the detection detector.
(f) After refining for a desired number of steps, the final predictions can be obtained.

References

[1] Swapnil Bhosale, Sauradip Nag, Diptesh Kanojia, Jiankang Deng, and Xiatian Zhu. Diffsed: Sound event detection with denoising diffusion. *arXiv preprint arXiv:2308.07293*, 2023. 3

[2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 3

[3] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection. *arXiv preprint arXiv:2211.09788*, 2022. 2, 3

[4] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 1

[5] Zheng Ge, Songtao Liu, Zeming Li, Osamu Yoshie, and Jian Sun. Ota: Optimal transport assignment for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 303–312, 2021. 2

[6] Z Ge, S Liu, F Wang, Z Li, and J Sun. Yolox: Exceeding yolo series in 2021. arxiv. *arXiv preprint arXiv:2107.08430*, 2021. 2

[7] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 1

[8] Matthieu Lin, Chuming Li, Xingyuan Bu, Ming Sun, Chen Lin, Junjie Yan, Wanli Ouyang, and Zhidong Deng. Detr for crowd pedestrian detection, 2021. 2

[9] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 2

[10] Sauradip Nag, Xiatian Zhu, Jiankang Deng, Yi-Zhe Song, and Tao Xiang. Difftad: Temporal action detection with proposal denoising diffusion. *arXiv preprint arXiv:2303.14863*, 2023. 2

[11] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 1

[12] Dingfeng Shi, Yujie Zhong, Qiong Cao, Jing Zhang, Lin Ma, Jia Li, and Dacheng Tao. React: Temporal action detection with relational queries. In *European conference on computer vision*, 2022. 2

[13] Jing Tan, Jiaqi Tang, Limin Wang, and Gangshan Wu. Relaxed transformer decoders for direct action proposal generation. In *ICCV*, 2021. 2

[14] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022. 3

[15] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, pages 6450–6459, 2018. 3

[16] Chen-Lin Zhang, Jianxin Wu, and Yin Li. Actionformer: Localizing moments of actions with transformers. In *European Conference on Computer Vision*, pages 492–510. Springer, 2022. 2