

# DiffTAD: Temporal Action Detection with Proposal Denoising Diffusion

Sauradip Nag<sup>1,2</sup> \*    Xiatian Zhu<sup>1,3</sup>    Jiankang Deng<sup>4</sup>    Yi-Zhe Song<sup>1,2</sup>    Tao Xiang<sup>1,2</sup>

<sup>1</sup> CVSSP, University of Surrey, UK    <sup>2</sup> iFlyTek-Surrey Joint Research Center on Artificial Intelligence, UK

<sup>3</sup> Surrey Institute for People-Centred Artificial Intelligence, UK    <sup>4</sup> Imperial College London, UK

## Abstract

We propose a new formulation of temporal action detection (TAD) with denoising diffusion, **DiffTAD** in short. Taking as input random temporal proposals, it can yield action proposals accurately given an untrimmed long video. This presents a generative modeling perspective, against previous discriminative learning manners. This capability is achieved by first diffusing the ground-truth proposals to random ones (i.e., the forward/noising process) and then learning to reverse the noising process (i.e., the backward/denoising process). Concretely, we establish the denoising process in the Transformer decoder (e.g., DETR) by introducing a temporal location query design with faster convergence in training. We further propose a cross-step selective conditioning algorithm for inference acceleration. Extensive evaluations on ActivityNet and THUMOS show that our DiffTAD achieves top performance compared to previous art alternatives. The code is available at <https://github.com/sauradip/DiffusionTAD>.

## 1. Introduction

Temporal action detection (TAD) aims to predict the temporal duration (i.e., start and end time) and the class label of each action instance in an untrimmed video [32, 9]. Existing methods rely on *proposal prediction* by regressing anchor proposals [86, 13, 21, 47] or predicting the start/end times of proposals [41, 8, 42, 90, 52, 87, 88, 50, 49, 51]. These models are all discriminative learning based.

In the generative learning perspective, diffusion model [66] has been recently exploited in image based object detection [15]. This represents a new direction for designing detection models in general. Although conceptually similar to object detection, the TAD problem presents more complexity due to the presence of temporal dynamics. Besides, there are several limitations with the detection diffusion formulation in [15]. First, a two-stage pipeline (e.g., RCNN [13]) is adopted, suffering localization-error prop-

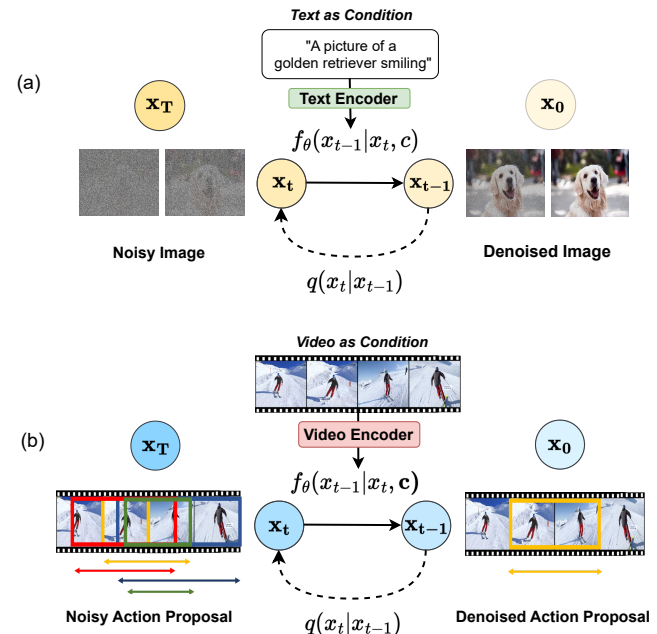


Figure 1. **Diffusion for temporal action detection (TAD)**. (a) A diffusion model for text-to-image generation where text embeddings are passed as the condition in the denoising (reverse) process. We draw an analogy by exploiting (b) a diffusion model for TAD: *To generate action temporal boundaries from noisy proposals with the condition of video embedding.*

agation from proposal generation to proposal classification [48]. Second, as each proposal is processed individually, their relationship modeling is overlooked, potentially hurting the learning efficacy. To avoid these issues, we adopt the one-stage detection pipelines [75, 80] that have already shown excellent performance with a relatively simpler design, in particular, DETR [11].

Nonetheless, it is non-trivial to integrate denoising diffusion with existing detection models, due to several reasons. (1) A unique challenge with TAD is **big boundary ambiguity**. This is because activities are continuous in time without clear start/end points (i.e., non-zero momentum), and the transition between consecutive activities is often stochastic. Further, human perception of action boundaries is also in-

\*This work was done during internship with Jiankang Deng.

stinctive and subjective. (2) Denoising diffusion and action detection both suffer low efficiency, and their combination would make it even worse. Both of the problems have not been investigated systematically thus far.

To address the aforementioned challenges, a novel **Proposal Denoising Diffusion** method is proposed for efficiently tackling the TAD task in a diffusion formulation, abbreviated as **DiffTAD**. It takes a set of random temporal proposals (*i.e.*, the start and end time pairs) following Gaussian distribution, and outputs the action proposals of a given untrimmed video. At training time, Gaussian noises are first added to the ground truth proposals to make *noisy* proposals. These discrete noisy proposals are then projected into a continuous vector space using sinusoidal projection [44] to form noisy queries in which the decoder (*e.g.*, DETR) will conduct the denoising diffusion process. Our denoising space choice facilitates the adoption of existing diffusion models, as discussed above. As a byproduct, the denoising queries strategy itself can accelerate the training convergence of DETR type models [38]. At inference time, conditioned on a test video, DiffTAD can generate action temporal boundaries by reversing the learned diffusion process from Gaussian random proposals. For improving inference efficiency, we further introduce a cross-timestep selective conditioning mechanism with two key functions: (1) minimizing the redundancy of intermediate predictions at each sampling step by filtering out the proposals far away from the distribution of corrupted proposals generated in training, and (2) conditioning the next sampling step by selected proposals to regulate the diffusion direction for more accurate inference.

Our **contributions** are summarized as follows. (1) For the first time we formulate the temporal action detection problem through denoising diffusion in the elegant transformer decoder framework. Additionally, integrating denoising diffusion with this decoder design solves the typical slow-convergence limitation. (2) We further enhance the diffusion sampling efficiency and accuracy by introducing a novel selective conditioning mechanism during inference. (3) Extensive experiments on ActivityNet and THUMOS benchmarks show that our DiffTAD achieves favorable performance against prior art alternatives.

## 2. Related Works

**Temporal action detection.** Inspired by object detection in static images [58], R-C3D [86] uses anchor proposals by following the design of proposal generation and classification. With a similar model design, TURN [21] aggregates local features to represent snippet-level features for temporal boundary regression and classification. SSN [96] decomposes an action instance into three stages (starting, course, and ending) and employs structured temporal pyramid pooling to generate proposals. BSN [42] pre-

dicts the start, end, and actionness at each temporal location and generates proposals with high start and end probabilities. The actionness was further improved in BMN [41] via additionally generating a boundary-matching confidence map for improved proposal generation. GTAN [47] improves the proposal feature pooling procedure with a learnable Gaussian kernel for weighted averaging. G-TAD [90] learns semantic and temporal context via graph convolutional networks for more accurate proposal generation. BSN++ [71] further extends BMN with a complementary boundary generator to capture rich context. CSA [70] enriches the proposal temporal context via attention transfer. VSGN [95] improves short-action localization using a cross-scale multi-level pyramidal architecture. Recently, Actionformer [93] and React [63] proposed a purely DETR based design for temporal action localization at multiple scales. Our DiffTAD is the very first TAD model which proposes action detection as a generative task.

**Diffusion models.** As a class of deep generative models, diffusion models [28, 67, 69] start from the sample in random distribution and recover the data sample via a gradual denoising process. Diffusion models have recently demonstrated remarkable results in fields including computer vision [5, 57, 61, 54, 26, 20, 60, 64, 27, 94, 29, 92], nature language processing [4, 39, 24], audio processing [55, 91, 85, 37, 73, 31, 35], interdisciplinary applications [33, 30, 3, 89, 76, 84, 62], *etc.* More applications of diffusion models can be found in recent surveys [92, 10].

**Diffusion model for perception tasks.** While Diffusion models have achieved great success in image generation [28, 69, 18], their potential for other perception tasks has yet to be fully explored. Some pioneer works tried to adopt the diffusion model for image segmentation tasks [82, 6, 25, 34, 7, 2, 16]. For example, Chen *et al.* [16] adopted Bit Diffusion model [17] for panoptic segmentation [36] of images and videos. However, despite significant interest in this idea, there is no previous solution that successfully adapts generative diffusion models for object detection, the progress of which remarkably lags behind that of segmentation. This might be because the segmentation task can be processed in an image-to-image style, which is more similar to image generation in formulation [14]. While object detection is a set prediction problem [11] with a need for assigning object candidates [59, 43, 11] to ground truth objects. However, Chen *et al.* [15] managed to apply a diffusion model to object detection for the first time. Similarly, we make the first attempt at formulating TAD in the diffusion framework by integrating the denoising process with the single-stage DETR architecture.

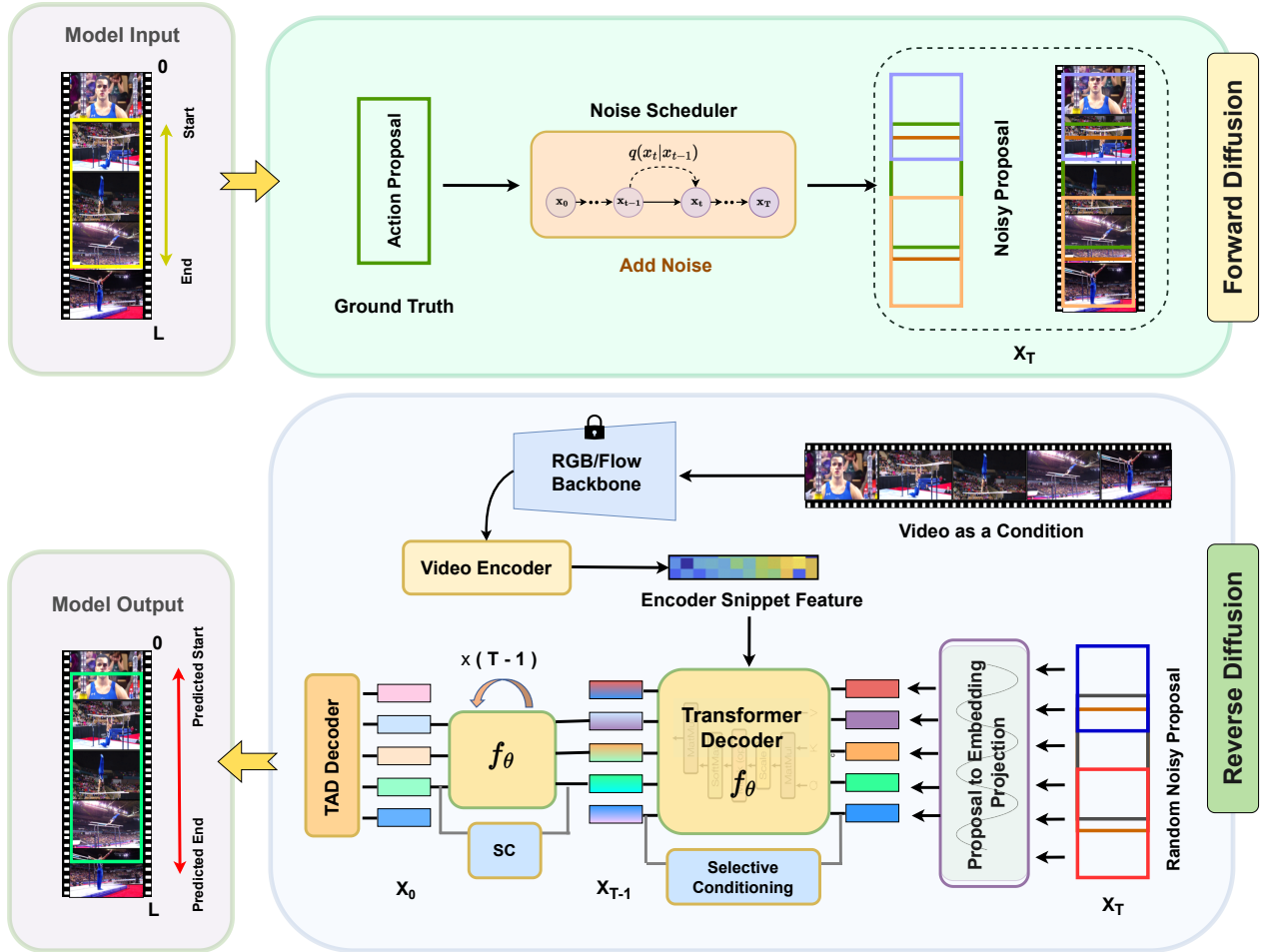


Figure 2. **Overview of our proposed DiffTAD.** In the forward diffusion process, Gaussian noises are added to the ground-truth boundaries iteratively to obtain noisy versions  $X_T$ . In the reverse denoising process, a video is passed as the condition along with random proposals sampled from Gaussian distribution. The discrete proposals are then projected to a continuous embedding space where proposal denoising takes place in an iterative fashion to obtain action proposals. In particular, a cross-timestep selective conditioning strategy is introduced for proposal refinement and filtering for more accurate and efficient inference.

### 3. Methodology

#### 3.1. Preliminaries

**Temporal action detection.** Our DiffTAD model takes as input an untrimmed video  $V$  with a variable number of frames. Video frames are first pre-processed by a feature encoder (*e.g.*, a Kinetics pre-trained I3D network [12]) into a sequence of localized snippets following the standard practice [41]. To train the model, we collect a set of labeled video training set  $\mathcal{D}^{train} = \{V_i, \Psi_i\}$ . Each video  $V_i$  is labeled with temporal annotation  $\Psi_i = \{(\psi_j, \xi_j, y_j)\}_{j=1}^{M_i}$  where  $\psi_j/\xi_j$  denote the start/end time,  $y_j$  is the action category, and  $M_i$  is the number of action instances.

**Diffusion models** [65, 28, 67, 66] are a class of likelihood-based models inspired by nonequilibrium thermodynamics [67, 68]. These models define a Markovian chain of diffusion forward process by gradually adding noises to the

sample data. The forward noising process is defined as

$$q(z_t|z_0) = \mathcal{N}(z_t|\sqrt{\bar{\alpha}_t}z_0, (1 - \bar{\alpha}_t)\mathbf{I}), \quad (1)$$

which transforms a sample  $z_0$  to a latent noisy sample  $z_t$  ( $t \in \{0, 1, \dots, T\}$ ) by adding noises to  $z_0$ .  $\bar{\alpha}_t := \prod_{s=0}^t \alpha_s = \prod_{s=0}^t (1 - \beta_s)$  and  $\beta_s$  represents the noise variance schedule [28]. During training, a neural network  $f_\theta(z_t, t)$  is trained to predict  $z_0$  from  $z_t$  by minimizing the training objective with  $\ell_2$  loss [28]:

$$\mathcal{L}_{train} = \frac{1}{2} \|f_\theta(z_t, t) - z_0\|^2. \quad (2)$$

At inference, a sample  $z_0$  is reconstructed from noise  $z_T$  with the model  $f_\theta$  and an updating rule [28, 66] in an iterative way, *i.e.*,  $z_T \rightarrow z_{T-\Delta} \rightarrow \dots \rightarrow z_0$ . More details of diffusion models can be found in Supplementary.

### 3.2. DiffTAD

**Diffusion-based TAD formulation.** In this work, we formulate the temporal action detection task in a conditional denoising diffusion framework. In our setting, data samples are a set of action temporal boundaries  $z_0 = \mathbf{b}$ , where  $\mathbf{b} \in \mathbb{R}^{N \times 2}$  denotes  $N$  temporal proposals. A neural network  $f_\theta(z_t, t, \mathbf{x})$  is trained to predict  $z_0$  from noisy proposals  $z_t$ , conditioned on the corresponding video  $V$ . The corresponding category label  $\hat{y}$  is produced accordingly.

Since the diffusion model generates a data sample iteratively, it needs to run the model  $f_\theta$  multiple times in inference. However, it would be computationally intractable to directly apply  $f_\theta$  on the raw video at every iterative step. For efficiency, we propose to separate the whole model into two parts, *video encoder* and *detection decoder*, where the former runs only once to extract a feature representation of the input video  $V_i$ , and the latter takes this feature as a condition to progressively refine the noisy proposals  $z_t$ .

**Video encoder.** The video encoder takes as input the pre-extracted video features and extracts high-level features for the following detection decoder. In general, any video encoder can be used. We use a video encoder same as [93]. More specifically, the raw video  $V$  is first encoded by a convolution encoder to obtain multiscale feature  $H_i \in \mathbb{R}^{T \times D}$  for RGB and optical flow separately. This is followed by a multi-scale temporal transformer [77]  $\mathcal{T}$  that performs global attention across the time dimension to obtain the global feature as:

$$F_g^i = \mathcal{T}(H_i), i \in [1, \dots, L] \quad (3)$$

where query/key/value of the transformer is set to  $H_i$  and  $L$  is the number of scales. We estimate the shared global representations  $F_g^i \in \mathbb{R}^{T \times D}$  across all the scales and concatenate them as  $F_g = \{F_g^0, F_g^1, \dots, F_g^L\}$ .

Previous TAD works [41, 90] typically use fused RGB and flow features (*i.e.*, early fusion) for modeling. Considering the feature specificity (RGB for appearance, and optical flow for motion), we instead exploit a late fusion strategy (Fig. 3). Specifically, we extract the video features  $F_g^{rgb}$  and  $F_g^{flow}$  for RGB and optical flow separately. The proposal denoising is then conducted in each space individually. The forward/noising process is random (*i.e.*, feature agnostic) and thus shared by both features. We will compare the two fusion strategies empirically (see Table 6).

**Detection decoder.** Similar to DETR [40], we use a transformer decoder [77] (denoted by  $f_\theta$ ) for detection. Functionally it serves as a denoiser. In traditional DETR, the queries are learnable continuous embeddings with random initialization. In DiffTAD, however, we exploit the queries as the *denoising targets*.

Specifically, we first project discrete proposals  $\psi \in$

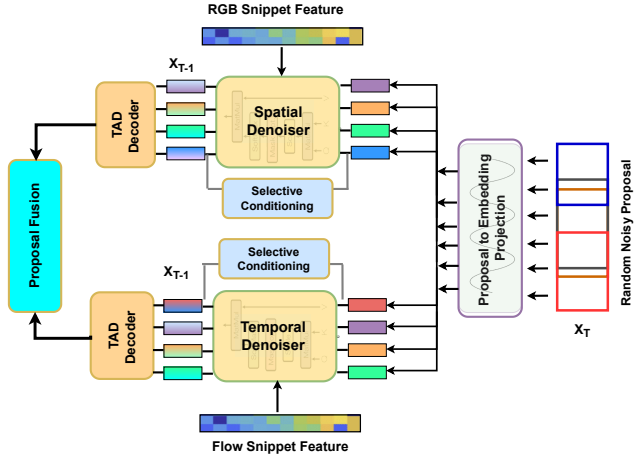


Figure 3. **Spacetime decoupled denoising.** We perform spacetime decoupled denoising via feature late-fusion. The RGB and optical flow features are extracted separately and passed as a condition to the denoiser, separately.

$\mathbb{R}^{N \times 2}$  to continuous query embeddings [81]:

$$Q = g(\psi) \in \mathbb{R}^{N \times D} \quad (4)$$

where  $g$  is MLP-based learnable projection. Taking  $Q$  as input, the decoder then predicts  $N$  outputs:

$$F_d = f_\theta(Q; F_g) \in \mathbb{R}^{N \times D} \quad (5)$$

where  $F_g$  is the global encoder feature and the  $F_d$  is the final embedding.  $F_d$  is finally decoded using three parallel heads namely (1) *action classification head*, (2) *action localization head*, and (3) *completeness head*, respectively. The first estimates the probability of a particular action within the action proposal. The second estimates the start, end and IOU overlap between the proposals. The third estimates the quality of predicted action proposals.

**Cross-timestep selective conditioning.** In DiffTAD, the denoising decoder  $f_\theta$  takes  $N$  action queries and then denoises each of them iteratively. Processing a large number of queries is thus inefficient. An intuitive way for better efficiency is to use static thresholds to suppress unimportant candidates [15], which however is ad-hoc and ineffective.

Here, we propose a more principled *cross-timestep selective conditioning* mechanism (Fig. 4). More specifically, we calculate a similarity matrix  $A \in \mathbb{R}^{N \times N}$  between the  $N$  queries  $x_t$  of current timestep and  $N$  queries  $x_{t+1}$  of conditioned/previous timestep  $x_{t+1}$ . Each element of  $A$  represents the similarity of the same queries across two successive timesteps. We select a set of queries according to:

$$\hat{P}_{sim} = \{(i, j) | A[i, j] - \gamma_1 > 0\} \quad (6)$$

where  $\gamma_1 \in [-1, 1]$  is a preset similarity threshold. We consider higher IoU with the desired (approximated by the



estimate of the last step) proposals, more effective for the queries to be denoised. Thus, we construct an IoU based matrix  $B \in \mathbb{R}^{N \times N}$  between two successive timesteps:

$$\hat{P}_{iou} = \{(i, j) | B[i, j] - \gamma_2 > 0\} \quad (7)$$

where  $i/j$  indexes the queries and  $\gamma_2 \in [-1, 1]$  is a preset IOU threshold. This allows for the most useful queries to be selected (see Fig. 1 in Supplementary).

We obtain the final query set as  $Q_c = (\hat{P}_{iou} / \hat{P}_{sim}) \cup Q$  with  $/$  denotes the set divide operation. For a selected query  $q_i$ , its key and value features can be obtained by fusion as  $K_i/V_i = \text{cat}(\{k_j/v_j | (i, j) \in Q_c\})$ . Each query feature is then updated as

$$\hat{q}_i = \text{softmax}(q_i K_i^\top) \cdot V_i^\top \quad (8)$$

These selected queries  $\{\hat{q}_i\}$  will be passed through the cross-attention with video embeddings for denoising. This selective conditioning is applied on a random proportion (e.g., 70%) of proposals per mini-batch during training, and on all the proposals during inference.

Our selective conditioning strategy shares the spirit of self-conditioning [17] in the sense that the output of the last step is used for accelerating the subsequent denoising. However, our design and problem setup both are different. For example, self-conditioning can be simply realized by per-sample feature concatenation, whilst our objective is to select a subset of queries based on the pairwise similarity and IoU measurement in an attention manner. We validate the conditioning design in the experiments (Table 2).

### 3.3. Training

During training, we first construct the diffusion process that corrupts the ground-truth proposals to noisy proposals. We then train the model to reverse this noising process. Please refer to Algorithm 1 for more details.

**Proposal corruption.** We add Gaussian noises to the ground truth action proposals. The noise scale is controlled by  $\alpha_t$  (in Eq. (1)), which adopts the monotonically decreasing cosine schedule in different timestep  $t$ , following [53]. Notably, the ground truth proposal coordinates need to be scaled as well since the signal-to-noise ratio has a significant effect on the performance of diffusion model [16]. We observe that TAD favors a relatively lower signal scaling value than object detection [15] (see Table 4). More discussions are given in Supplementary.

**Training losses.** The detection detector takes as input  $N_{train}$  corrupted proposals and predicts  $N_{train}$  predictions each including the category classification, proposal coordinates, and IOU regression. We apply the set prediction loss [11, 72, 97] on the set of  $N_{train}$  predictions. We assign multiple predictions to each ground truth by selecting the top  $k$  predictions with the least cost by an optimal transport assignment method [22, 23, 83, 19].

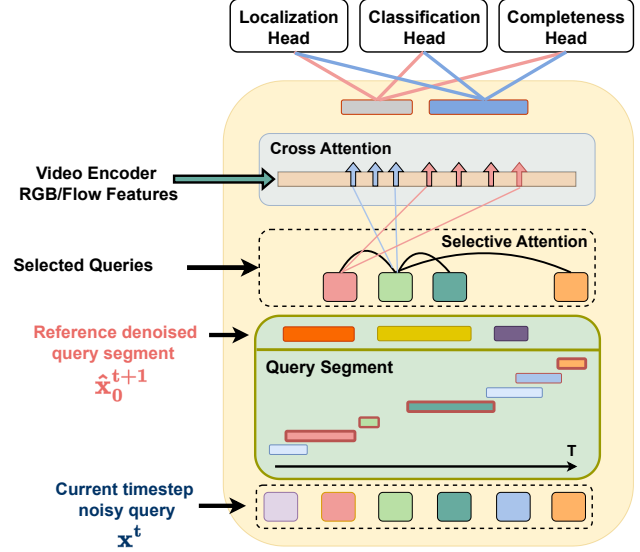


Figure 4. **Cross-timestep selective conditioning.** At the current time step, noisy query only attends to other queries selectively based on the overlap and similarity with the previously denoised reference proposals/segments. For clarity, LayerNorm, FFN, and residual connection are omitted.

---

#### Algorithm 1 DiffTAD Training

---

```
def train(video_feat, gt_proposals):
    # Encode image features
    feats = video_encoder(video_feat)

    # Signal scaling
    pb = (pb * 2 - 1) * scale

    # Corrupt gt_proposals
    t = randint(0, T) # time step
    eps = normal(mean=0, std=1) # noise: [B, N, 2]

    pb_crpt = sqrt(alpha_cumprod(t)) * pb +
              sqrt(1 - alpha_cumprod(t)) * eps

    # Project to continuous embedding
    pb_crpt = project(pb_crpt) # query : [B, N, D]

    # Calculate Self-condition estimate
    pb_pred = zeros_like(pb_crpt)
    if self_cond and uniform(0,1) > 0.7:
        pb_pred = decoder(pb_crpt, pb_pred, feats, t)
        pb_pred = stop_gradient(pb_pred)

    # Predict
    pb_pred = decoder(pb_crpt, pb_pred, feats, t)

    # Set prediction loss
    loss = set_prediction_loss(pb_pred, gt_proposals)

    return loss
```

---

$\text{alpha\_cumprod}(t)$ : cumulative product of  $\alpha_i$ , i.e.,  $\prod_{i=1}^t \alpha_i$

### 3.4. Inference

In inference, starting from noisy proposals sampled in Gaussian distribution, the model progressively refines the predictions as illustrated in Algorithm 2.

**Sampling step.** At each sampling step, the random or es-

---

**Algorithm 2** DiffTAD Sampling

---

```
def infer(video_feat, steps, T):  
    # Encode video features  
    feats = video_encoder(video_feat)  
  
    # noisy proposals: [B, N, 2]  
    pb_t = normal(mean=0, std=1)  
  
    # noisy embeddings: [B, N, D]  
    pb_t = project(pb_t)  
  
    pb_pred = zeros_like(pb_t)  
  
    # uniform sample step size  
    times = reversed(linespace(-1, T, steps))  
  
    # [(T-1, T-2), (T-2, T-3), ..., (1, 0), (0, -1)]  
    time_pairs = list(zip(times[:-1], times[1:]))  
  
    for t_now, t_next in zip(time_pairs):  
        # Predict pb_0 from pb_t  
        if not self_cond:  
            pb_pred = zeros_like(pb_t)  
            pb_pred = decoder(pb_t, pb_pred, feats, t_now)  
  
        # Estimate pb_t at t_next  
        pb_t = ddim_step(pb_t, pb_pred, t_now, t_next)  
  
    return pb_pred
```

---

linespace: generate evenly spaced values

timated proposals from the last sampling step are first projected into the continuous query embeddings and sent into the detection decoder to predict the category, proposal IOU and proposal coordinates. After obtaining the proposals of the current step, DDIM [66] is adopted to estimate the proposals for the next step.

**Proposal prediction.** DiffTAD has a simple proposal generation pipeline without post-processing (e.g., non-maximum suppression). To make a reliable confidence estimation for each proposal, we fuse the action classification score  $p_{bc}$  and completeness score  $p_c$  for each proposal with a simple average to obtain the final proposal score  $p_{sc}$ .

### 3.5. Key Insights

**One model multiple trade-offs.** Once trained, DiffTAD works under multiple settings with a varying number of proposals and sampling steps during inference (Fig. 5(b)). In general, better accuracy can be obtained using more proposals and more steps. Thus, a single DiffTAD can realize a number of trade-off needs between speed and accuracy.

**Faster convergence.** DETR variants suffer generally slow convergence [45] due to two reasons. First, inconsistent updates of the anchors, the objective for the object queries to learn, would make the optimization of target boundaries difficult. Second, the ground truth assignment using a dynamic process (e.g., Hungarian matching) is unstable due to both the nature of discrete bipartite matching and the stochastic training process. For instance, a small perturbation with the cost matrix might cause enormous matching and inconsis-

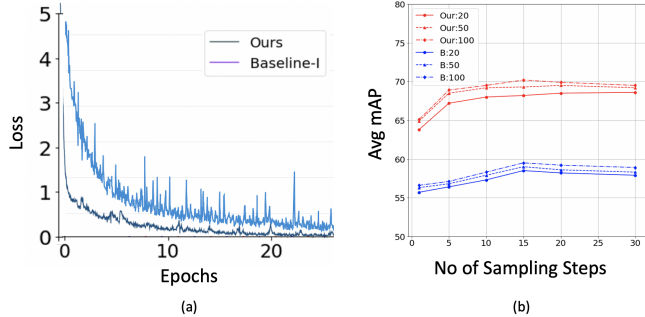


Figure 5. **Properties of DiffTAD.** (a) Our proposed model converges more stably during training. (b) Its performance increases with the number of queries (20/50/100) and also with the sampling steps, with a clear margin over the Baseline (B-20/50/100). Dataset: THUMOS.

tent optimization. Our DiffTAD takes a denoising strategy that makes learning easier. More specifically, each query is designed as a proposal proxy, a noised query, that can be regarded as a good anchor due to being close to a ground truth. With the ground truth proposal as a definite optimization objective, the ambiguity brought by Hungarian matching can be well suppressed. We validate that our query denoising based DiffTAD converges more stably than DiffusionDet [15] based Baseline-I (Fig. 5(a)), whilst achieving superior performance (Table 1).

**Better sampling.** We evaluate DiffTAD under a varying number of (10/50/100) random proposals by increasing the sampling steps from 1 to 30. As shown in Table 1, under all three settings, DiffTAD yields steady performance gains from more steps consumed. In particular, in the case of fewer random proposals, often DiffTAD can achieve larger gains than DiffusionDet [15]. For example, in the case of 50 proposals, the mAP of DiffTAD boosts the avg mAP from 64.9% (1 step) to 68.5% (5 steps), i.e., an absolute gain of 3.6% avg mAP. Unlike object detection, we find TAD benefits little from increasing the number of proposals. One intuitive reason is that positive samples are less in TAD than in object detection. Compared to the previous two-stage refinement of discriminative learning based TAD models [74], our gain is also more decent (0.8% vs. 4.2% (10 steps)). This is because it lacks a principled iterative inference ability as in diffusion models.

## 4. Experiments

**Datasets.** We conduct extensive experiments on two popular TAD benchmarks. (1) *ActivityNet-v1.3* [9] has 19,994 videos from 200 action classes. We follow the standard setting to split all videos into training, validation and testing subsets in ratio of 2:1:1. (2) *THUMOS14* [32] has 200 validation videos and 213 testing videos from 20 categories with labeled temporal boundary and action class.

Table 1. Performance comparison with the state-of-the-art methods. Metrics: mAP at different IoU thresholds, and average mAP in [0.3 : 0.1 : 0.7] on THUMOS14 and [0.5 : 0.05 : 0.95] on ActivityNet-v1.3.

Models	Design	Feature	THUMOS						ActivityNet			
			0.3	0.4	0.5	0.6	0.7	Avg	0.5	0.75	0.95	Avg
<i>Discriminative learning based models</i>												
TAL-Net [13]	2-stage	I3D	53.2	48.5	42.8	33.8	20.8	-	38.2	18.3	1.3	20.2
BMN [41]	2-stage	TSN	56.0	47.4	38.8	29.7	20.5	38.5	50.1	34.8	8.3	33.9
GTAD [90]	2-stage	TSN	54.5	47.6	40.3	30.8	23.4	39.3	50.4	34.6	9.0	34.1
RTD-Net [74]	2-stage	I3D	68.3	62.3	51.9	38.8	23.7	-	47.2	30.7	8.6	30.8
TCANet [56]	2-stage	I3D	60.6	53.2	44.6	36.8	26.7	44.3	52.3	36.7	6.9	35.5
MUSES [46]	2-stage	I3D	68.9	64.0	56.9	46.3	31.0	—	50.0	35.0	6.6	34.0
ContextLoc [98]	2-stage	I3D	68.3	63.8	54.3	41.8	26.2	50.9	56.0	35.2	3.6	34.2
RCL [79]	2-stage	I3D	70.1	62.3	52.9	42.7	30.7	57.1	51.7	35.2	8.0	34.4
React [63]	1-stage	I3D	69.2	65.0	57.1	47.8	35.6	55.0	49.6	33.0	8.6	32.6
TAGS [48]	1-stage	I3D	68.6	63.8	57.0	46.3	31.8	52.8	<b>56.3</b>	36.8	<b>9.6</b>	<b>36.5</b>
ActionFormer [93]	1-stage	I3D	82.1	77.8	71.0	59.4	43.9	66.8	53.5	36.2	8.2	35.6
<i>Generative learning based models</i>												
Baseline(1-step)	2-stage	I3D	65.2	61.3	55.4	44.6	35.5	52.4	48.5	31.4	8.6	31.5
Baseline(5-step)	2-stage	I3D	69.1	65.7	60.2	47.1	36.4	55.7	50.2	32.3	8.9	32.2
Baseline(10-step)	2-stage	I3D	70.0	66.5	60.6	47.5	36.9	56.3	51.0	32.9	9.0	32.4
<b>DiffTAD(1-step)</b>	1-stage	I3D	68.7	66.8	64.7	<b>61.2</b>	<b>57.3</b>	63.8	52.4	35.6	8.8	34.8
<b>DiffTAD(5-step)</b>	1-stage	I3D	73.4	71.5	69.9	<b>62.8</b>	<b>58.4</b>	<b>67.2</b>	55.2	<b>36.8</b>	8.9	36.0
<b>DiffTAD(10-step)</b>	1-stage	I3D	74.9	72.8	<b>71.2</b>	<b>62.9</b>	<b>58.5</b>	<b>68.0</b>	56.1	<b>36.9</b>	9.0	36.1

## 4.1. Implementation details

**Training schedule.** For video feature extraction, we use Kinetics pre-trained I3D model [12, 93] with a downsampling ratio of 4 and R(2+1)D model [1, 93] with a downsampling ratio of 2. Our model is trained for 50 epochs using Adam optimizer with a learning rate of  $10^{-4}/10^{-5}$  for ActivityNet/THUMOS, respectively. The batch size is set to 40 for ActivityNet and 32 for THUMOS. The values of  $\gamma_1/\gamma_2$  are kept as 0.7/0.5 respectively. For selective conditioning, we apply the rate of 70% during training. All models are trained with 4 NVIDIA-V100 GPUs. Refer to Supplementary for additional implementation details.

**Testing schedule.** At the inference stage, the detection decoder iteratively refines the predictions from Gaussian random proposals. By default, we set the total sampling time steps as 10.

## 4.2. Main results

**Competitors.** We compare our DiffTAD with the state-of-the-art non-generative approaches including BMN [41], GTAD [90], React [63] and ActionFormer [93]. Further, we adapt the object detection counterpart DiffusionDet [15] to a two-stage generative TAD method, termed as *Baseline*.

**Results on THUMOS.** We make several observations from Table 1: (1) Our DiffTAD achieves the best result, surpassing strong discriminative learning based competitors like TCANet [78] and ActionFormer [93] by a clear mar-

gin. This suggests the overall performance advantage of our model design and generative formulation idea. (2) Importantly, DiffTAD excels on the alternative generative learning design (*i.e.*, *Baseline*), validating the superiority of our diffusion-based detection formulation. For both generative models, more sampling steps lead to higher performance. This concurs with the general property of diffusion models. (3) In particular, our model achieves significantly stronger results in stricter IOU metrics (*e.g.*, IOU@0.5/0.6/0.7), as compared to all the other alternatives. This demonstrates the potential of generative learning in tackling the action boundary often with high ambiguity, and the significance of proper diffusion design.

**Results on ActivityNet.** Similar observations can be drawn in general on ActivityNet from Table 1. We further highlight several differences: (1) Indeed, overall our DiffTAD is not the best performer, with a slight edge underneath TAGS [48]. However, we note that all other DETR style methods (*e.g.*, RTD-Net) are significantly inferior. This means that our method has already successfully filled up the most performance disadvantage of the DETR family. We attribute this result to our design choice of denoising in the query space and cross-timestep selective conditioning. That being said, our formulation in exploiting the DETR architecture for TAD is superior than all prior attempts. (2) Compared to the generative competitor (*i.e.*, *Baseline*), our model is not only more stable to converge (Fig. 5(a)), but also yields a margin of 3.7% (smaller than that on THUMOS as this is

Table 2. **Proposal refinement design.** Dataset: THUMOS.

Design	mAP			
	0.3	0.5	0.7	Avg
Feature Concatenation [17]	71.1	66.3	56.5	65.9
Proposal Renewal [15]	71.2	65.4	54.2	65.1
<b>Selective Condition (Ours)</b>	<b>74.9</b>	<b>71.2</b>	<b>58.5</b>	<b>68.0</b>

a more challenging test for the DETR family in general).

### 4.3. Ablation study

We conduct ablation experiments on THUMOS to study DiffTAD in detail. In all experiments, we use 30 proposals for training and inference, unless specified otherwise.

**Cross-timestep selective conditioning.** We examine the effect of the proposed selective conditioning for proposal refinement (Section 3.2). To that end, we vary the rate/portion of proposals per batch at which selective conditioning is applied during training. The case of 0% training rate means *no conditioning*. Note, during inference selective conditioning is always applied to all the proposals (*i.e.*, 100% test rate). As demonstrated in Fig. 6(b), we observe a clear correlation between conditioning rate and sampling quality (*i.e.*, mAP), validating the importance of our selective conditioning in refining proposals for enhanced denoising.

Additionally, we compare with two different refinement designs: (1) *Feature concatenation* as in self-conditioning [17], and (2) *Proposal renewal* as in DiffusionDet [15]. We observe from Table 2 that our selective conditioning is superior to both alternatives, verifying our design idea.

**Sampling decomposition.** We decompose the sampling strategy with DiffTAD by testing four variants: (1) *No denoising*: No DDIM [66] is applied, which means the output prediction of the current step is used directly as the input of the next step (*i.e.*, a naive baseline). (2) *Vanilla denoising*: DDIM is applied in the original form. (3) *Selective conditioning*: Only selective conditioning is used but no vanilla denoising. (4) *Both vanilla denoising and selective conditioning*: Our full model. We observe from Table 3 that (1) Without proper denoising, applying more steps will cause more degradation, as expected, because there is no optimized Markovian chain. (2) Applying vanilla denoising can address the above problem and improve the performance from multiple sampling accordingly. However, the performance increase saturates quickly. (3) Interestingly, our proposed selective conditioning even turns out to be more effective than vanilla denoising. (4) When both vanilla denoising and selective conditioning are applied (*i.e.*, our full model), the best performances can be achieved, with a clear gain over the either and even improved saturation phenomenon. This suggests that the two ingredients are largely complementary, which is not surprising given their distinc-

Table 3. **Sampling decomposition** in terms of iterative denoising (ID) and selective conditioning (SC). Dataset: THUMOS.

ID	SC	step 1	step 5	step 10
✗	✗	62.7	62.9	61.3
✓	✗	62.7	65.0	64.9
✗	✓	62.7	65.7	65.6
✓	✓	<b>62.7</b>	<b>67.2</b>	<b>68.0</b>

Table 4. **Signal-noise ratio** under a variety of different scaling factors. Dataset: THUMOS.

Scale	mAP			
	0.3	0.5	0.7	Avg
0.1	70.5	67.5	57.2	66.6
<b>0.5</b>	<b>74.9</b>	<b>71.2</b>	<b>58.5</b>	<b>68.0</b>
1.0	73.2	70.6	58.0	67.3
2.0	69.9	67.3	56.8	64.2

Table 5. **Accuracy vs. speed** under a variety of different proposal numbers and sampling steps. Dataset: THUMOS.

Proposals	Step	mAP				GFLOPs
		0.3	0.5	0.7	Avg	
30	5	74.9	71.2	58.5	68.0	0.81
50	1	69.2	65.2	53.0	64.9	1.27
<b>50</b>	<b>5</b>	<b>77.2</b>	<b>73.5</b>	<b>59.1</b>	<b>68.5</b>	<b>3.17</b>
100	1	70.4	66.1	53.8	65.1	2.18
100	5	77.1	73.8	58.7	68.4	6.32

tive design nature.

**Signal scaling.** The signal scaling factor (Eq. (1)) controls the signal-to-noise ratio (SNR) of the denoising process. We study the influence of this facet in DiffTAD. As shown in Table 4, the scaling factor of 0.5 achieves the optimal performance. This means this setting is task-specific at large (*e.g.*, the best choice is 2.0 for object detection [15]).

**Accuracy vs. speed.** We evaluate the trade-off between the accuracy and memory cost with DiffTAD. In this test, we experiment with three choices {10, 50, 100} in the number of proposals. For each choice, the same is applied to both model training and evaluation consistently. We observe in Table 5 that: (1) Increasing the number of proposals/queries from 30 to 50 brings about 0.5% in mAP with extra 0.46 GFLOPs, thanks to the proposed selective conditioning. (2) However, further increasing the proposals is detrimental to the model performance. This is because the number of action instances per video is much fewer than that of objects per image on average. (3) With 50 proposals, increasing the sampling steps from 1 to 5 provides an mAP gain of 3.6% with an additional cost of 1.9 GFLOPs. A similar observation is with the case of 100 proposals.

**Decoupled diffusion strategy.** We evaluate the impact of feature decoupling (video encoder in Sec. 3.2). Typically, existing TAD methods [90, 41] use fused RGB and optical flow video features (*i.e.*, early fusion). However, we take



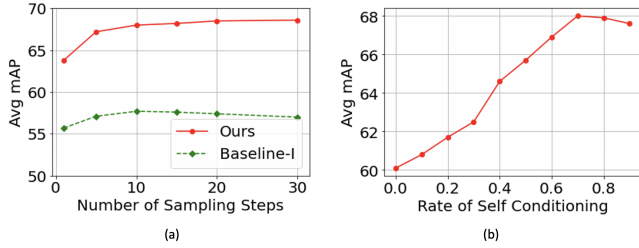


Figure 6. **Impact of sampling and selective-conditioning.** (a) The effect of varying sampling steps with Baseline and DiffTAD. (b) The effect of selective conditioning rate during training DiffTAD. Dataset: THUMOS.

Table 6. **Decoupling the denoising.** RGB and optical flow video features are decoupled for individual denoising (*i.e.*, late fusion), in contrast to typical early fusion strategy where the two features are first fused in prior to being processed. Dataset: THUMOS.

Decoupling	0.5	0.75	0.95	Avg
✗	71.8	67.4	57.1	66.0
✓	<b>74.9</b>	<b>71.2</b>	<b>58.5</b>	<b>68.0</b>

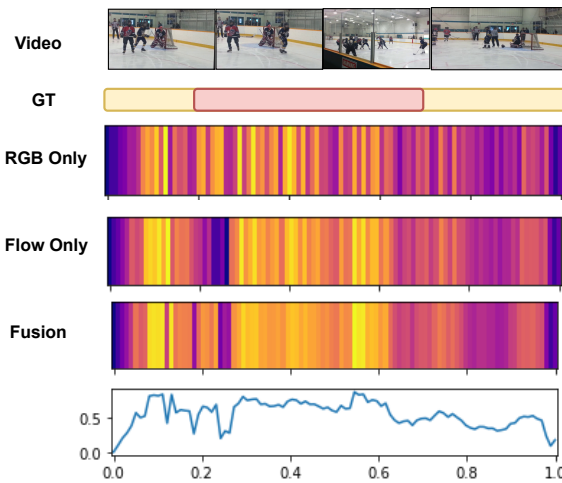


Figure 7. **Late fusion example** on a test video from THUMOS. It is observed that RGB and optical flow features are complementary useful in capturing the ground truth (GT) action instance.

a late fusion strategy where the RGB and flow features are processed separately before their proposals are fused (Fig. 3). We contrast the two fusion strategies. It is evident in Table 6 that with the typical early fusion (*i.e.*, passing early fused features as a condition to our detection decoder), a drop of 2% in average mAP is resulted. This indicates that due to modal specificity there is a need for specific conditional inference, validating our design consideration. For visual understanding, an example is given in Fig. 7 to show how the two features can contribute to TAD in a cooperative manner.

**NMS-free design.** As shown in Table 7, when comparing DiffTAD with and without non-maximum suppression

Table 7. Ablation study on non-maximum suppression (NMS) on THUMOS14, measured by AR@AN.

Model	AR@50	AR@100	AR@500
BMN [41]	29.0	37.7	56.1
BMN [41] + NMS	32.7	40.7	56.4
DiffTAD	63.6	69.6	<b>73.1</b>
DiffTAD + NMS	<b>64.3</b>	<b>69.9</b>	72.8

Table 8. Denoising strategy with DiffTAD on THUMOS14.

Denoising Strategy		mAP			
Proposal	Label	0.3	0.5	0.7	Avg
✗	✓	70.1	65.9	52.7	62.4
✓	✗	<b>74.9</b>	<b>71.2</b>	<b>58.5</b>	<b>68.0</b>
✓	✓	73.8	70.4	58.0	67.3

(NMS), we observe similar results. NMS is not necessary in DiffTAD because the predictions are relatively sparse and minor overlapped with our cross-step conditional strategy (Sec. 3.2). In contrast, existing non-generative TAD works like BMN [41] and GTAD [90] generate highly overlapped proposals with similar confidence. Thus, NMS becomes necessary to suppress redundant proposals.

**Ablation of denoising strategy.** Due to the inherent query based design with the detection decoder, (1) we can corrupt discrete action proposals and project them as queries, and (2) we can also corrupt the action label and project it as label queries. Both can be taken as the input to the decoder. For corrupting the label queries, we use random shuffle as the noise in the forward diffusion step. To validate this design choice experimentally, we test three variants: (1) only labels are corrupted, (2) only proposals are corrupted, and (3) both proposals and labels are corrupted. For the non-corrupted quantity, we add noise to the randomly initialized embedding. For the last variant, we stack all the corrupted proposals and labels and pass them into the decoder. It can be observed in Table 8 that corrupting labels alone observes the most drop in performance, and corrupting both labels and proposals is inferior to only corrupting the proposals.

## 5. Conclusion

In this work, we propose a novel temporal action detection (TAD) paradigm, DiffTAD, by considering it as a denoising diffusion process from noisy proposals to action proposals. This proposed generative model is conceptually distinctive from all previous TAD methods based on discriminative learning. Our model is designed by tailoring the diffusion and denoising process in a single-stage DETR framework, with appealing properties such as more stable convergence, flexible proposal sizes, and superior proposal refinement. Experiments on standard benchmarks show that DiffTAD achieves favorable performance compared to both generative and non-generative alternatives.

## References

- [1] Humam Alwassel, Silvio Giancola, and Bernard Ghanem. Tsp: Temporally-sensitive pretraining of video encoders for localization tasks. *arXiv*, 2020. 7
- [2] Tomer Amit, Eliya Nachmani, Tal Shaharbany, and Lior Wolf. Segdiff: Image segmentation with diffusion probabilistic models. *arXiv preprint arXiv:2112.00390*, 2021. 2
- [3] Namrata Anand and Tudor Achim. Protein structure and sequence generation with equivariant denoising diffusion probabilistic models. *arXiv preprint arXiv:2205.15019*, 2022. 2
- [4] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021. 2
- [5] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18208–18218, 2022. 2
- [6] Dmitry Baranchuk, Andrey Voynov, Ivan Rubachev, Valentin Khulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. In *International Conference on Learning Representations*, 2022. 2
- [7] Emmanuel Asiedu Brempong, Simon Kornblith, Ting Chen, Niki Parmar, Matthias Minderer, and Mohammad Norouzi. Denoising pretraining for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4175–4186, 2022. 2
- [8] Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles. Sst: Single-stream temporal action proposals. In *CVPR*, 2017. 1
- [9] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, pages 961–970, 2015. 1, 6
- [10] Hanqun Cao, Cheng Tan, Zhangyang Gao, Guangyong Chen, Pheng-Ann Heng, and Stan Z Li. A survey on generative diffusion model. *arXiv preprint arXiv:2209.02646*, 2022. 2
- [11] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 1, 2, 5
- [12] 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, 7
- [13] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A Ross, Jia Deng, and Rahul Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. In *CVPR*, 2018. 1, 7
- [14] Jiaqi Chen, Jiachen Lu, Xiatian Zhu, and Li Zhang. Generative semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 2
- [15] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusionet: Diffusion model for object detection. *arXiv preprint arXiv:2211.09788*, 2022. 1, 2, 4, 5, 6, 7, 8
- [16] Ting Chen, Lala Li, Saurabh Saxena, Geoffrey Hinton, and David J Fleet. A generalist framework for panoptic segmentation of images and videos. *arXiv preprint arXiv:2210.06366*, 2022. 2, 5
- [17] Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202*, 2022. 2, 5, 8
- [18] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 2
- [19] Yuming Du, Wen Guo, Yang Xiao, and Vincent Lepetit. 1st place solution for the uvo challenge on image-based open-world segmentation 2021. *arXiv preprint arXiv:2110.10239*, 2021. 5
- [20] Wanshu Fan, Yen-Chun Chen, Dongdong Chen, Yu Cheng, Lu Yuan, and Yu-Chiang Frank Wang. Frido: Feature pyramid diffusion for complex scene image synthesis. *ArXiv*, abs/2208.13753, 2022. 2
- [21] Jiyang Gao, Zhenheng Yang, Kan Chen, Chen Sun, and Ram Nevatia. Turn tap: Temporal unit regression network for temporal action proposals. In *ICCV*, 2017. 1, 2
- [22] 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. 5
- [23] Z Ge, S Liu, F Wang, Z Li, and J Sun. Yolox: Exceeding yolo series in 2021. *arxiv*. *arXiv preprint arXiv:2107.08430*, 2021. 5
- [24] Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and LingPeng Kong. Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv preprint arXiv:2210.08933*, 2022. 2
- [25] Alexandros Graikos, Nikolay Malkin, Nebojsa Jojic, and Dimitris Samaras. Diffusion models as plug-and-play priors. *arXiv preprint arXiv:2206.09012*, 2022. 2
- [26] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10696–10706, 2022. 2
- [27] William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. Flexible diffusion modeling of long videos. *arXiv preprint arXiv:2205.11495*, 2022. 2
- [28] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 2, 3
- [29] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv preprint arXiv:2204.03458*, 2022. 2
- [30] Emiel Hoogetboom, Victor Garcia Satorras, Clement Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. *arXiv e-prints*, pages arXiv–2203, 2022. 2

- [31] Rongjie Huang, Zhou Zhao, Huadai Liu, Jinglin Liu, Chenye Cui, and Yi Ren. Prodiff: Progressive fast diffusion model for high-quality text-to-speech. *arXiv preprint arXiv:2207.06389*, 2022. 2
- [32] Haroon Idrees, Amir R Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The thumos challenge on action recognition for videos “in the wild”. *Computer Vision and Image Understanding*, 155:1–23, 2017. 1, 6
- [33] Bowen Jing, Gabriele Corso, Regina Barzilay, and Tommi S Jaakkola. Torsional diffusion for molecular conformer generation. In *ICLR2022 Machine Learning for Drug Discovery*, 2022. 2
- [34] Boah Kim, Yujin Oh, and Jong Chul Ye. Diffusion adversarial representation learning for self-supervised vessel segmentation. *arXiv preprint arXiv:2209.14566*, 2022. 2
- [35] Sungwon Kim, Heeseung Kim, and Sungroh Yoon. Guided-tts 2: A diffusion model for high-quality adaptive text-to-speech with untranscribed data. *arXiv preprint arXiv:2205.15370*, 2022. 2
- [36] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019. 2
- [37] Alon Levkovitch, Eliya Nachmani, and Lior Wolf. Zero-shot voice conditioning for denoising diffusion tts models. *arXiv preprint arXiv:2206.02246*, 2022. 2
- [38] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13619–13627, 2022. 2
- [39] Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *arXiv preprint arXiv:2205.14217*, 2022. 2
- [40] Matthieu Lin, Chuming Li, Xingyuan Bu, Ming Sun, Chen Lin, Junjie Yan, Wanli Ouyang, and Zhidong Deng. Detr for crowd pedestrian detection, 2021. 4
- [41] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3889–3898, 2019. 1, 2, 3, 4, 7, 8, 9
- [42] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. BSN: Boundary sensitive network for temporal action proposal generation. In *ECCV*, 2018. 1, 2
- [43] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 2
- [44] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. Dab-detr: Dynamic anchor boxes are better queries for detr. *arXiv preprint arXiv:2201.12329*, 2022. 2
- [45] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. DAB-DETR: Dynamic anchor boxes are better queries for DETR. In *International Conference on Learning Representations*, 2022. 6
- [46] Xiaolong Liu, Yao Hu, Song Bai, Fei Ding, Xiang Bai, and Philip HS Torr. Multi-shot temporal event localization: a benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12596–12606, 2021. 7
- [47] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei. Gaussian temporal awareness networks for action localization. In *CVPR*, 2019. 1, 2
- [48] Sauradip Nag, Xiatian Zhu, Yi-zhe Song, and Tao Xiang. Proposal-free temporal action detection via global segmentation mask learning. In *ECCV*, 2022. 1, 7
- [49] Sauradip Nag, Xiatian Zhu, Yi-zhe Song, and Tao Xiang. Semi-supervised temporal action detection with proposal-free masking. In *ECCV*, 2022. 1
- [50] Sauradip Nag, Xiatian Zhu, Yi-zhe Song, and Tao Xiang. Zero-shot temporal action detection via vision-language prompting. In *ECCV*, 2022. 1
- [51] Sauradip Nag, Xiatian Zhu, Yi-zhe Song, and Tao Xiang. Post-processing temporal action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18837–18845, June 2023. 1
- [52] Sauradip Nag, Xiatian Zhu, and Tao Xiang. Few-shot temporal action localization with query adaptive transformer. *arXiv preprint arXiv:2110.10552*, 2021. 1
- [53] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 5
- [54] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 16784–16804. PMLR, 17–23 Jul 2022. 2
- [55] Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, and Mikhail Kudinov. Grad-tts: A diffusion probabilistic model for text-to-speech. In *ICML*, 2021. 2
- [56] Zhiwu Qing, Haisheng Su, Weihao Gan, Dongliang Wang, Wei Wu, Xiang Wang, Yu Qiao, Junjie Yan, Changxin Gao, and Nong Sang. Temporal context aggregation network for temporal action proposal refinement. In *CVPR*, pages 485–494, 2021. 7
- [57] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *ArXiv*, abs/2204.06125, 2022. 2
- [58] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *TPAMI*, 39(6):1137–1149, 2016. 2
- [59] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149, 2017. 2

- [60] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. *ArXiv*, abs/2208.12242, 2022. 2
- [61] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. 2
- [62] Arne Schneuing, Yuanqi Du, Charles Harris, Arian Jamasb, Ilia Igashov, Weitao Du, Tom Blundell, Pietro Lió, Carla Gomes, Max Welling, Michael Bronstein, and Bruno Correia. Structure-based drug design with equivariant diffusion models. *arXiv preprint arXiv:2210.13695*, 2022. 2
- [63] 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, 7
- [64] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 2
- [65] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 3
- [66] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 1, 3, 6, 8
- [67] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019. 2, 3
- [68] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020. 3
- [69] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 2
- [70] Deepak Sridhar, Niamul Quader, Srikanth Muralidharan, Yaoxin Li, Peng Dai, and Juwei Lu. Class semantics-based attention for action detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13739–13748, 2021. 2
- [71] Haisheng Su, Weihao Gan, Wei Wu, Yu Qiao, and Junjie Yan. Bsn++: Complementary boundary regressor with scale-balanced relation modeling for temporal action proposal generation. *arXiv preprint arXiv:2009.07641*, 2020. 2
- [72] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14454–14463, 2021. 5
- [73] Jaesung Tae, Hyeongju Kim, and Taesu Kim. Editts: Score-based editing for controllable text-to-speech. *arXiv preprint arXiv:2110.02584*, 2021. 2
- [74] Jing Tan, Jiaqi Tang, Limin Wang, and Gangshan Wu. Relaxed transformer decoders for direct action proposal generation. In *ICCV*, 2021. 6, 7
- [75] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019. 1
- [76] Brian L Trippe, Jason Yim, Doug Tischer, Tamara Broderick, David Baker, Regina Barzilay, and Tommi Jaakkola. Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. *arXiv preprint arXiv:2206.04119*, 2022. 2
- [77] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 4
- [78] Lining Wang, Haosen Yang, Wenhao Wu, Hongxun Yao, and Hujie Huang. Temporal action proposal generation with transformers. *arXiv preprint arXiv:2105.12043*, 2021. 7
- [79] Qiang Wang, Yanhao Zhang, Yun Zheng, and Pan Pan. Rcl: Recurrent continuous localization for temporal action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13566–13575, 2022. 7
- [80] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. Solo: Segmenting objects by locations. In *ECCV*, 2020. 1
- [81] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based object detection. *arXiv preprint arXiv:2109.07107*, 3(6), 2021. 4
- [82] Julia Wolleb, Robin Sandkühler, Florentin Bieder, Philippe Valmaggia, and Philippe C Cattin. Diffusion models for implicit image segmentation ensembles. *arXiv preprint arXiv:2112.03145*, 2021. 2
- [83] Junfeng Wu, Qihao Liu, Yi Jiang, Song Bai, Alan Yuille, and Xiang Bai. In defense of online models for video instance segmentation. *arXiv preprint arXiv:2207.10661*, 2022. 5
- [84] Lemeng Wu, Chengyue Gong, Xingchao Liu, Mao Ye, and Qiang Liu. Diffusion-based molecule generation with informative prior bridges. *arXiv preprint arXiv:2209.00865*, 2022. 2
- [85] Shoule Wu and Ziqiang Shi. Itôts and itôwave: Linear stochastic differential equation is all you need for audio generation. *arXiv e-prints*, pages arXiv–2105, 2021. 2
- [86] Huijuan Xu, Abir Das, and Kate Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *ICCV*, 2017. 1, 2
- [87] Mengmeng Xu, Juan-Manuel Pérez-Rúa, Victor Escorcía, Brais Martínez, Xiatian Zhu, Li Zhang, Bernard Ghanem, and Tao Xiang. Boundary-sensitive pre-training for temporal localization in videos. In *ICCV*, pages 7220–7230, 2021. 1
- [88] Mengmeng Xu, Juan-Manuel Perez-Rua, Xiatian Zhu, Bernard Ghanem, and Brais Martínez. Low-fidelity end-to-



- end video encoder pre-training for temporal action localization. In *NeurIPS*, 2021. 1
- [89] Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. In *International Conference on Learning Representations*, 2021. 2
- [90] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. G-tad: Sub-graph localization for temporal action detection. In *CVPR*, 2020. 1, 2, 4, 7, 8, 9
- [91] Dongchao Yang, Jianwei Yu, Helin Wang, Wen Wang, Chao Weng, Yuexian Zou, and Dong Yu. Diffsound: Discrete diffusion model for text-to-sound generation. *arXiv preprint arXiv:2207.09983*, 2022. 2
- [92] Ruihan Yang, Prakhar Srivastava, and Stephan Mandt. Diffusion probabilistic modeling for video generation. *arXiv preprint arXiv:2203.09481*, 2022. 2
- [93] 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, 4, 7
- [94] Mingyuan Zhang, Zhongang Cai, Liang Pan, Fangzhou Hong, Xinying Guo, Lei Yang, and Ziwei Liu. Motiondiffuse: Text-driven human motion generation with diffusion model. *arXiv preprint arXiv:2208.15001*, 2022. 2
- [95] Chen Zhao, Ali K Thabet, and Bernard Ghanem. Video self-stitching graph network for temporal action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13658–13667, 2021. 2
- [96] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *ICCV*, 2017. 2
- [97] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable {detr}: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021. 5
- [98] Zixin Zhu, Wei Tang, Le Wang, Nanning Zheng, and Gang Hua. Enriching local and global contexts for temporal action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13516–13525, 2021. 7