

Action Detection via an Image Diffusion Process

Lin Geng Foo¹ Tianjiao Li¹ Hossein Rahmani² Jun Liu^{1†}

¹Singapore University of Technology and Design ²Lancaster University

{lingeng_foo,tianjiao_li}@mymail.sutd.edu.sg,

h.rahmani@lancaster.ac.uk, jun_liu@sutd.edu.sg

Abstract

Action detection aims to localize the starting and ending points of action instances in untrimmed videos, and predict the classes of those instances. In this paper, we make the observation that the outputs of the action detection task can be formulated as images. Thus, from a novel perspective, we tackle action detection via a three-image generation process to generate starting point, ending point and action-class predictions as images via our proposed Action Detection Image Diffusion (ADI-Diff) framework. Furthermore, since our images differ from natural images and exhibit special properties, we further explore a Discrete Action-Detection Diffusion Process and a Row-Column Transformer design to better handle their processing. Our ADI-Diff framework achieves state-of-the-art results on two widely-used datasets.

1. Introduction

The goal of action detection is to localize the starting and ending points of action instances in untrimmed videos, while also predicting the classes of those actions. Action detection is important across many video analysis applications, including healthcare monitoring [44, 47], sports analysis [17, 24] and security surveillance [1, 60], and has attracted a lot of research attention. A common approach [10, 16, 29, 62, 63] is to first extract proposals of action instances, before processing each of these proposals individually to produce refined starting point, ending point, and action-class predictions. Many works focus on improving the action proposal localization process (e.g., with a better starting and ending point regression head [30, 31, 36]), or designing better model architectures (e.g., graph models [26, 65, 70] and Transformers [35, 69]). Nevertheless, action detection still remains challenging, since actions often contain complex motions with high intra-class variability [21, 73], and difficulties also arise due to the varying lighting conditions, different viewpoints and background clutter [21, 29, 48].

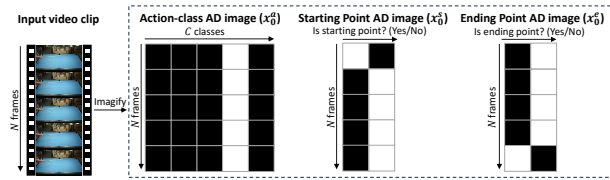


Figure 1. Illustration of our formulated AD images, which allow us to tackle action detection by generating three images. The action-class AD image (x^a) has a shape of $N \times C$, while the starting and ending point AD images (x^s and x^e) both have a shape of $N \times 2$, where we show $N = 5$ and $C = 5$ in this figure for illustration. Specifically, the pixel values in a row of the image form the probabilities of a discrete distribution regarding a specific video frame, e.g., the n -th row of the action-class AD image represents the probability distribution over the action classes for the n -th frame. We depict the ground truth AD images (x_0^a, x_0^s, x_0^e) in this figure, thus each row contains a single white pixel (with value 1) in each row depicting the correct prediction, while the other pixels are black in color (with value 0).

On the other hand, image diffusion models [22, 56] have recently undergone rapid development and show an excellent capability to generate high-quality images. Image diffusion models aim to obtain a high-quality image from a noisy and uncertain image, and achieve this via step-by-step progressive denoising. Intuitively, the diffusion model’s process of progressive denoising helps to bridge the large gap between the high-quality target images and the noisy images by breaking it down into smaller intermediate steps [55], which assists the model in converging towards generating the high-quality target images. Thus, image diffusion models [22, 56] can improve image quality and training stability, and possess a strong ability to generate high-quality target images that align well with the provided input conditions.

In this work, inspired by the efficacy of image diffusion models, we make the following observation: the three outputs (starting point, ending point and action-class) for the action detection task can be *formulated as images*. For instance, the action-class predictions can be represented by a $N \times C$ image (where N is the number of frames and C is the number of action-classes), while the starting and ending point predictions can each be represented by a $N \times 2$ image,

† Corresponding author

as shown in Fig. 1. Hence, from a new perspective, we can re-cast action detection as a three-image generation task, tackled by generating these three “action detection” images – which we call *AD images* – as output. Then, in order to generate these AD images with a high level of quality, we can leverage image diffusion models [22, 53, 54] with their strong image generation capabilities.

To this end, we propose an AD Image Diffusion (ADI-Diff) framework for action detection, as shown in Fig. 2. Our ADI-Diff framework learns to generate the target high-quality action-class, starting point and ending point AD images via diffusion. Following previous works on diffusion models [22, 54], our ADI-Diff framework comprises two opposite diffusion processes: the *forward process* and the *reverse process*. Specifically, the forward process aims to generate supervisory signals of intermediate steps during training, through progressively adding noise to the ground truth AD images. Conversely, the reverse process aims to learn to reverse the forward process, i.e., by learning to denoise and produce high-quality AD images, which is the main part of our action detection pipeline.

However, directly using standard diffusion models [22, 54] for our ADI-Diff framework can be sub-optimal since they learn to generate natural images, while our proposed AD images differ from natural images, because AD images also represent a set of *discrete probability distributions*. For instance, our AD images are used to tackle a classification problem (either among C action-classes or 2 classes for starting/ending point predictions), which is a problem of predicting discrete probability distributions. Thus, our AD images also represent a set of discrete probability distributions, where the pixels in each row of the image represent the probabilities of a discrete distribution. Hence, instead of following the standard diffusion process to map between a high-quality image and a totally uncertain image, our diffusion process should learn to map between the ground truth – which is an *ideal discrete probability distribution* – and a *totally uncertain discrete probability distribution*. In other words, the standard diffusion process, which progressively introduces Gaussian noise in the forward process and converges towards Gaussian noise, is not well-suited for our needs. Therefore, we propose a novel Discrete Action-Detection Diffusion Process that constrains each forward diffusion step to produce discrete probability distributions, which enables us to generate the desired high-quality AD images from the noisy and uncertain probability distributions more effectively.

Moreover, in contrast to traditional images which contain rich local spatial correlations in both dimensions, our AD images exhibit different relationship patterns along each of the two dimensions. Specifically, in our AD images, there is a strong sequential ordering between adjacent rows (i.e., between temporal frames), which differs from the inter-class relationships between adjacent columns (e.g., between action

classes). Hence, since our AD images differ from traditional images, existing diffusion network designs, which tend to focus on 2D spatial processing in local neighbourhoods, are not suitable for our use. Thus, we further propose our Row-Column Transformer design for our diffusion model to effectively extract class information across the columns while encoding temporal relationships across the rows.

In summary, our contributions are as follows: **(1)** From a novel perspective, we re-cast action detection as a three-image generation problem and generate the AD image predictions via our AD Image Diffusion (ADI-Diff) framework. **(2)** We propose a Discrete Action-Detection Diffusion Process that constrains the forward diffusion process to produce discrete probability distributions, which provides a good mapping between the input noisy distribution and the ground truth distribution. **(3)** To handle our AD images which are different from traditional images, we further introduce a Row-Column Transformer design for our diffusion network.

2. Related Work

In **action detection**, many approaches [10, 16, 29, 62, 63] first extract action proposals before processing them individually to predict action classes and refined starting and ending points. These methods are generally split into two categories: anchor-based and anchor-free. Anchor-based methods [10, 16, 63] such as multi-tower networks [9] and temporal feature pyramid networks [33, 36] generate a dense set of anchors with pre-defined lengths throughout the video that act as action proposals. On the other hand, anchor-free methods [29, 40, 42, 62, 69] often predict actionness scores [61, 72] or action boundary confidence scores [29, 30] for video frames in order to generate action proposals. Besides, some existing works explore different architectures to encode spatio-temporal information, including RNNs [5, 67], graph models [3, 26, 65, 68, 70], or Transformers [8, 35, 43, 58]. Moreover, some proposal-free methods [40, 41] have also been explored recently, and our ADI-Diff also falls into this category. Different from previous works, we re-cast action detection as a three-image generation problem, and leverage the diffusion model’s strong image generation capability to generate the three AD image predictions. Our proposed ADI-Diff framework effectively handles the challenging action detection task by generating high-quality starting/ending point and action-class AD images, achieving good results.

Diffusion models have emerged as an effective way to sample from a data distribution by learning to estimate the gradients of the data distribution [55]. Originally introduced in the context of image generation [53], diffusion models have seen much development in recent years [2, 15, 22, 54, 64], and have been explored across various generation tasks, including video [52], point cloud [39] and text [27] generation. Diffusion models have also been adopted for human activity analysis [14, 20, 32, 49], e.g., for

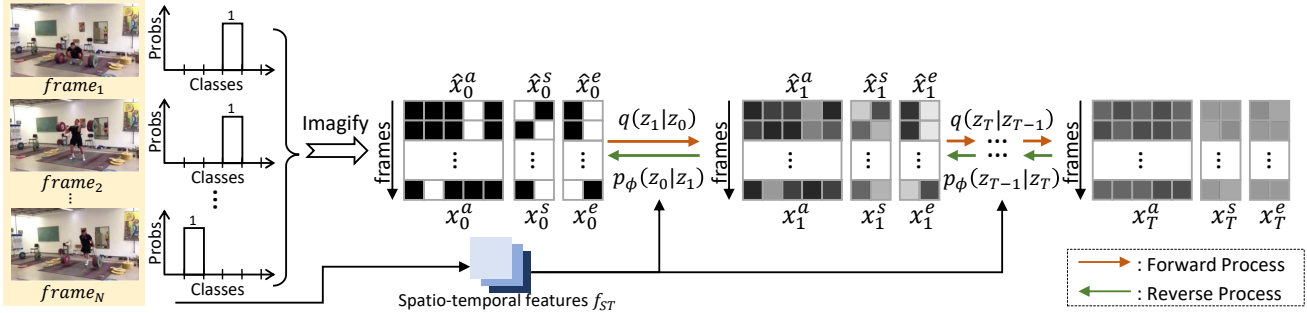


Figure 2. Illustration of the proposed AD Image Diffusion (ADI-Diff) framework. The forward process (represented with orange arrows) progressively diffuses the ground truth AD images x_0^a, x_0^s, x_0^e towards a noisy outcome, which generates supervisory signals for intermediate steps. On the other hand, the reverse process (represented with green arrows) is trained to denoise the noisy inputs x_T^a, x_T^s, x_T^e while conditioned on extracted spatio-temporal features f_{ST} from the input video, to obtain the output AD images $\hat{x}_0^a, \hat{x}_0^s, \hat{x}_0^e$.

pose estimation [20] where a GMM-based forward process is proposed. Moreover, some works [11] adopt diffusion models for regressing the bounding box of an object, which has also inspired a similar process for regressing temporal boundaries [42]. In this work, considering that action detection requires starting point, ending point and action-class outputs which can be treated as three AD images, and that diffusion models naturally have a strong image generation capability, we propose to cast action detection as a three-image diffusion process. At the same time, since AD images are not natural images and have their own properties, we propose modifications to the image diffusion process, which attain good performance.

3. ADI-Diff Framework

In this paper, we tackle the action detection task by *casting it as an image generation problem* (as described in Sec. 3.1), and leverage a diffusion model via our proposed ADI-Diff framework to generate AD images that encode the required starting point, ending point and action-class information. Moreover, in Sec. 3.2, we design a Discrete Action-Detection Diffusion Process, constraining the pixels along each row of our AD images to form a discrete probability distribution in the forward process. Besides, in order to perform diffusion for our AD images which exhibit different relationship patterns along each of the two dimensions (i.e., between frames and between classes), we propose a Row-Column Transformer architecture in Sec. 3.3.

3.1. Formulation of AD images

In this work, we observe that we can reformulate the three outputs of the action detection task (starting point, ending point and action-class prediction) as three images. Hence, from a new perspective, we can cast action detection as an image diffusion process to generate these three AD images. Below, we describe how we formulate our action-class, starting point and ending point AD images to encode the corresponding predictions.

Action-class AD image x^a . As shown in Fig. 1, to capture the action-class predictions at each time step, we set the action-class AD image to be a matrix x^a with shape $N \times C$, where N is the number of frames in the video and C represents the number of action classes. The matrix x^a can be seen as an image, where the pixel value at the n -th row and c -th column is the probability that action c is happening at the n -th frame of the video, and is constrained to be in $[0, 1]$. Thus, x^a is a grayscale image with shape $N \times C$, which can be generated via an image diffusion process.

Starting Point and Ending Point AD images x^s, x^e . Next, in order to encode temporal boundary predictions, we produce two AD images: the starting point AD image $x^s \in [0, 1]^{N \times 2}$ and the ending point AD image $x^e \in [0, 1]^{N \times 2}$, which respectively encode predictions of the starting points and ending points of action instances. Specifically, the N rows of the matrix x^s (or x^e) encode information regarding the N frames, with the 2 pixels in each row respectively encoding the probabilities for the presence and absence of a starting (or ending) point. For example, for the starting point AD image x^s , the pixel value in the first column of the n -th row represents the probability of a starting point occurring at the n -th frame. The same goes for the ending point AD image x^e , except that the pixel value in the first column encodes the probability of an ending point occurring instead. In summary, both x^s and x^e can be seen as grayscale images with shape $N \times 2$, as shown in Fig. 1.

3.2. Discrete Action-Detection Diffusion Process

After re-casting action detection as a three-image generation task, we seek to generate high-quality AD images to handle action detection effectively. To achieve this, we derive inspiration from the strong image generation capabilities of diffusion models [22, 54], and adopt a diffusion-based approach to generate the three AD images for action detection. Overall, diffusion models [22, 54] aim to *obtain a high-quality image from a totally noisy and uncertain image*, and do so by progressively removing the noise and uncertainty over multiple steps. Specifically, to learn a mapping

between a noisy image (that is totally random and uncertain) and a high-quality image, standard diffusion models consist of a *forward process* where Gaussian noise is progressively added to high-quality images. Meanwhile, the *reverse process* learns to reverse the forward process, i.e., to denoise the noisy and uncertain inputs to obtain high-quality images. These processes enable diffusion models to bridge the large gap between the input noisy images and the target high-quality images, and *obtain high-quality images from noisy and uncertain images*.

Nevertheless, using standard image-based diffusion models [22, 54] directly can be sub-optimal. Specifically, these diffusion models aim to generate natural images from noisy images, so they add Gaussian noise during the forward process to naturally obtain intermediate noisy images. However, our AD images differ from natural images, since we are using our AD images to deal with a classification problem, e.g., the action-class AD images are used to tackle a C -way action classification task, while the starting point AD images are used to predict starting points as a binary classification task (yes/no), and the same goes for ending point AD images. In other words, our AD images are in fact a set of *discrete probability distributions*. Thus, using the Gaussian noise is not suitable, because we want our diffusion process to learn to map between the ground truth – an ideal *discrete probability distribution* – and a totally uncertain *discrete probability distribution*, with intermediate discrete distributions to bridge the gap. In order to form intermediate discrete probability distributions, we cannot simply apply Gaussian noise during the forward process, instead we need to constrain each step of the forward process to produce a *discrete probability distribution*, and also converge towards a totally uncertain *discrete probability distribution*. Hence, below we design our own Discrete Action-Detection Diffusion Process.

Firstly, following the standard diffusion model that adds random noise during their forward process to obtain a totally noisy and uncertain image, we would like to add random noise to obtain a totally noisy and uncertain discrete distribution in our forward diffusion process. We note that, when the classification predictions are totally uncertain, there should be an equal probability of predicting any class, which corresponds to the *Uniform distribution*. Since the Uniform distribution is the most uncertain, we would like to add noise to converge towards the Uniform distribution in our forward process (**Property 1**).

Apart from fulfilling Property 1 above, we find that previous diffusion models [22] also have two other important properties that help to facilitate the training of the step-by-step diffusion process. Therefore, to maintain the efficacy of the diffusion framework, we here also need to satisfy these two properties, which are as follows: **Property 2**) Following previous diffusion models [22], there should be a formula to efficiently jump over t forward steps, i.e., a formulation

for $q(z_t|z_0)$ (as in Eq. 4). This formula facilitates training and allows us to randomly sample multiple time steps, without having to iterate through many forward steps to get to a specified step. **Property 3**) Following previous diffusion models [22], we also need a formulation for the forward process posterior, i.e., $q(z_{t-1}|z_t, z_0)$ in Eq. 5, which allows us to generate z_{t-1} from z_t to form a z_{t-1}, z_t pair, enabling a direct step-wise comparison against the reverse process step during training [22].

Therefore, in order to fulfill these properties, and learn to generate the three AD images via our diffusion, we design our forward and reverse process as described below.

Forward Process. In the forward diffusion process, we aim to create the supervisory signals of intermediate steps for training. Notably, the forward process of previous works add Gaussian noise at every forward step, which eventually corrupts a natural image (at step 0) into Gaussian noise (at step T). Here, we instead wish to add a specific type of noise to fulfill Property 1. Thus, we initialize the ground truth AD images, and then progressively diffuse (the rows of) the ground truth AD images towards the Uniform distribution over T steps.

Next, we formally introduce some definitions. For simplicity, we describe the diffusion process for a single row of the action-class AD image as an example, which is a discrete probability distribution with C classes. Specifically, we define z_t to be a discrete probability distribution at step t of the diffusion process, where z_t is a vector of length C . At step 0 of the diffusion process, z_0 represents the ground truth, and is a one-hot vector with value 1 at the index of the ground truth category, and 0 elsewhere. The forward process spans T steps, where noise is gradually added to z_0 , such that after T steps, z_T is approximately a Uniform distribution.

Concretely, to create intermediate distributions $\{z_1, \dots, z_T\}$ from z_0 , we add random noise v_t at each t -th forward step, which progressively makes the prediction more uncertain. This addition of random noises ($\{v_t\}_{t=1}^T$) is an important component [22, 55, 56] that facilitates exploration of the low-density regions of the data distribution. Here, we add v_t at each step according to a *Multinomial distribution* with uniform probability parameters, which crucially allows us to *converge towards the Uniform distribution* and satisfy Property 1 (as explained in further detail later).

Specifically, at each step t , to obtain z_t from z_{t-1} , we increase the uncertainty by introducing a small chance to randomly select any class (which might not be the correct ground truth class) with equal probability. Here, we introduce β_t as a small positive hyperparameter to control the small increase in randomness at step t . Following the above intuition, we can formulate each t -th forward step as:

$$z_t = (1 - \beta_t)z_{t-1} + \beta_t v_t, \quad (1)$$

where v_t is a random vector of length C whose elements are non-negative and add up to 1, i.e., forming the probabilities

of a discrete distribution. We obtain v_t via sampling from a $MN_K(K, \frac{1}{C}\mathbf{1})$ distribution, where MN stands for the Multinomial distribution, K is a hyperparameter for the number of trials, $\frac{1}{C}\mathbf{1}$ gives a uniform probability of selecting each class in each trial (where $\mathbf{1}$ is a vector of 1's with length C), and we further divide the resulting sample by K to let elements of v_t sum to 1 (denoted by the subscript K).

Therefore, the likelihood $q(z_t|z_{t-1})$ of observing z_t given z_{t-1} can be formulated as:

$$q(z_t|z_{t-1}) = MN_{\frac{K(z_t - (1-\beta_t)z_{t-1})}{\beta_t}}(z_t; K, \frac{1}{C}\mathbf{1}, z_{t-1}), \quad (2)$$

where, with slight abuse of notation, $MN(z_t; \cdot)$ is the Multinomial's likelihood of observing z_t , and the subscript is the substitution formula (Kv_t in terms of z_t) which is used to formulate the exact likelihood, such that Kv_t follows the Multinomial distribution (more details in Supp).

Next, expanding upon Eq. 1 which represents a single forward step, the formula for t steps of the forward process starting from z_0 can be derived as:

$$z_t = \bar{\alpha}_t z_0 + \left(\prod_{\tau=2}^t \alpha_\tau \right) \beta_1 v_1 + \left(\prod_{\tau=3}^t \alpha_\tau \right) \beta_2 v_2 + \dots + \beta_t v_t, \quad (3)$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{\tau=1}^t \alpha_\tau$. Then, the corresponding likelihood $q(z_t|z_0)$ of observing z_t (after t steps) can be approximately formulated as the following (with the proof and analysis in Supp):

$$q(z_t|z_0) = MN_{\frac{B_t K(z_t - \bar{\alpha}_t z_0)}{1 - \bar{\alpha}_t}}(z_t; B_t K, \frac{1}{C}\mathbf{1}, z_0), \quad (4)$$

where $B_t = \frac{(1 - \bar{\alpha}_t)^2}{((\prod_{\tau=2}^t \alpha_\tau)^2 \beta_1^2 + (\prod_{\tau=3}^t \alpha_\tau)^2 \beta_2^2 + \dots + \beta_t^2)}$.

Note that, we can show that our diffusion process *fulfills Property 1*. Specifically, when we set our β_t 's to be relatively high such that $\bar{\alpha}_t$ converges to 0 as $t \rightarrow T$, the likelihood in Eq. 4 converges towards $q(z_T|z_0) = MN_{B_T K}(B_T K, \frac{1}{C}\mathbf{1})$. Then, using the properties of the Multinomial distribution, we observe that z_T approximately converges to a Uniform distribution in expectation, i.e., $\mathbb{E}[z_T] = \frac{1}{C}\mathbf{1}$. This explicitly shows that our diffusion framework satisfies Property 1. At the same time, we also *fulfill Property 2*, since Eq. 4 enables us to directly generate the intermediate distributions $\{z_1, \dots, z_T\}$ from z_0 for efficient training.

Next, we would like to fulfill Property 3, which will provide a way to obtain z_{t-1} from z_t in the forward process, giving us a pair of z_{t-1}, z_t to facilitate the step-wise training of the reverse process [2, 22, 54]. Specifically, this requires a formulation of the forward process posterior $q(z_{t-1}|z_t, z_0)$. We can formulate a tractable expression for $q(z_{t-1}|z_t, z_0)$ by using the properties of the Markov chain, as follows:

$$q(z_{t-1}|z_t, z_0) = \frac{1}{\sigma_t} (MN_{\frac{K(z_t - (1-\beta_t)z_{t-1})}{\beta_t}}(z_{t-1}; K, \frac{1}{C}\mathbf{1}, z_t)) \cdot (MN_{\frac{B_{t-1} K(z_{t-1} - \bar{\alpha}_{t-1} z_0)}{1 - \bar{\alpha}_{t-1}}}(z_{t-1}; B_{t-1} K, \frac{1}{C}\mathbf{1}, z_0)), \quad (5)$$

where $\sigma_t = \sum_{z_{t-1}} [(MN_{\frac{K(z_t - (1-\beta_t)z_{t-1})}{\beta_t}}(z_{t-1}; K, \frac{1}{C}\mathbf{1}, z_t)) \cdot$

$(MN_{\frac{B_{t-1} K(z_{t-1} - \bar{\alpha}_{t-1} z_0)}{1 - \bar{\alpha}_{t-1}}}(z_{t-1}; B_{t-1} K, \frac{1}{C}\mathbf{1}, z_0))]$ (more details in Supp). Note that, in practice we can fix σ_t as a hyperparameter, since it is constant for all observed z_{t-1} .

Reverse Process. Using the forward process presented above, we can generate the intermediate distributions $\{z_1, \dots, z_T\}$. Then, we can use these intermediate distributions to optimize our diffusion model d (parameterized by ϕ) to learn the reverse diffusion process. As shown in Fig. 2, the reverse process aims to generate the AD image outputs after T diffusion steps.

First, we extract information from the input video to facilitate the diffusion process. To do this, we follow previous works [29, 40, 50, 51, 60, 69] and extract features from video snippets with a pre-trained feature extractor. Specifically, we extract a feature $f_{ST} \in \mathbb{R}^{N \times C_{ST}}$, where N is the number of frames and C_{ST} is the number of channels. Each reverse step will be conditioned on this extracted feature f_{ST} .

Next, we perform the reverse diffusion process. We first initialize the input noisy distribution z_T that is approximately a Uniform distribution – more precisely, z_T follows a $MN_{B_T K}(B_T K, \frac{1}{C}\mathbf{1})$ distribution to match with the z_T of our forward process. Then, we perform the reverse diffusion process $z_T \rightarrow \hat{z}_{T-1} \rightarrow \dots \rightarrow \hat{z}_0$ to generate the predictions \hat{z}_0 , where the hat ($\hat{\cdot}$) operator denotes that these are estimates produced by our diffusion model d (and not the forward process). Specifically, we define our reverse process in a step-by-step manner as follows:

$$\hat{z}_{t-1} = d_\phi(\hat{z}_t, f_{ST}, f_t), \quad t \in \{1, \dots, T\}, \quad (6)$$

where f_t is the unique step embedding to represent the t^{th} diffusion step, which we generate via the sinusoidal function. By performing the T steps of the reverse process via Eq. 6, we can produce output predictions \hat{z}_0 from the uncertain distribution z_T . Moreover, to better represent the intermediate and target distributions, we initialize M samples during training and perform the reverse process on M samples, instead of using a single sample only.

Multi-row Processing. For simplicity, above we describe the forward and reverse process in terms of a *single row* z_t of the action-class AD image. However, the same diffusion process can be simultaneously applied to all the rows of the action-class AD image x^a , where the forward process adds noise to all rows at once (to generate $\{x_1^a, \dots, x_T^a\}$ from the ground truth x_0^a), and the reverse process aims to reverse the addition of noise in all rows (i.e., produce x_{t-1}^a from x_t^a at the t -th step). We further note that this Discrete Action-Detection Diffusion Process is also used for the starting point and ending point AD images, since they all tackle a frame-wise classification task.

3.3. Row-Column Transformer Architecture

Moreover, different from natural images which contain rich local 2D spatial relationships, our AD images possess different relationship patterns along each of the two dimensions

(i.e., between frames vs. between classes). Specifically, there is a strong natural sequential ordering between adjacent rows (i.e., between adjacent temporal frames), which yet differs from the inter-class relationships between adjacent columns (e.g., between adjacent action classes). Hence, existing diffusion network designs [22, 54], which tend to focus on 2D spatial processing in neighbourhoods, are not well-suited for our use. To overcome this, we propose to handle the two dimensions in different ways, such that our Row-Column Transformer design can effectively extract class information across the columns while encoding temporal information across the rows. Below, for simplicity, we describe the architecture of our diffusion network d to handle a single AD image, using the action-class AD image as an example.

Next, we describe the inputs to the diffusion network d at the t -th diffusion step. We denote the input image as $x^a \in \mathbb{R}^{N \times C}$. In order to derive predictions specific to the input video, we also extract spatio-temporal features $f_{ST} \in \mathbb{R}^{N \times C_{ST}}$ from the input video. By conditioning on f_{ST} , our reverse process receives important video-specific information to perform the denoising. Besides, to better capture the distribution characteristics at each t -th step, we also condition the diffusion process on step index t , and generate a diffusion step embedding $f_t \in \mathbb{R}^{N \times 1}$ via the sinusoidal function to represent the t -th diffusion step. Then, we concatenate x^a , f_{ST} and f_t to form input $x \in \mathbb{R}^{N \times (C + C_{ST} + 1)}$.

Our Row-Column Transformer design for our diffusion network d consists of L stacks of *Row-Column Blocks*, which we introduce below. Refer to Supp for more details.

Row-Column Block. The first part of the block encodes information across columns, i.e., class information. Crucially, relationships between the columns (i.e., inter-class relationships) can exist over long ranges. Hence, in order to encode the relationships between columns (classes) across a long range, we perform a Multi-Head Self-Attention (MHSA) between the columns of the input image. Note that, for starting and ending point AD images, the two neighbouring columns (yes/no) are highly correlated, and these correlations can still be learned with the diffusion design in the previous section and the MHSA operation. Specifically, we treat each column of the input $x \in \mathbb{R}^{N \times (C + C_{ST} + 1)}$ as a token, thus obtaining $C + C_{ST} + 1$ tokens of length N . Next, a learnable positional embedding is added to each token, which encodes the positional information of each token. Then, we perform MHSA among all the $C + C_{ST} + 1$ tokens, to obtain an intermediate output $u_{col} \in \mathbb{R}^{N \times (C + C_{ST} + 1)}$. This is followed by 2 MLP layers, where we eventually output $x_{col} \in \mathbb{R}^{N \times (C + C_{ST} + 1)}$.

In the next part of the Row-Column Block, we encode temporal information across rows (frames). Notably, there exist *strong local relationships and sequential correlations* between neighbouring rows (frames). Furthermore, actions often provide context information for other actions in the

same sequence, which we can exploit by considering the *longer-term temporal relationships* between rows (frames). Thus, to effectively encode local relationships between neighbouring rows (frames), we apply a Temporal Convolution (TC), which has a strong inductive bias for encoding local sequential relationships [12, 13]. We also combine the TC with a MHSA conducted between the rows (frames) to encode long-range temporal relationships. Specifically, we first process the local relationships with a 1×3 TC, to yield $u_{row} \in \mathbb{R}^{N \times (C + C_{ST} + 1)}$. Then, to encode long-range temporal relationships, we first add a learnable positional embedding to each token, before performing MHSA across the rows (frames) by treating each row of u_{row} as a token (i.e., there are N tokens of length $C + C_{ST} + 1$). This is followed by 2 MLP layers to obtain a final output $x_{row} \in \mathbb{R}^{N \times (C + C_{ST} + 1)}$.

Combined Image Processing. Above, we describe both the Discrete Action-Detection Diffusion Process and Row-Column Transformer for the action-class AD image x^a . Yet, these methods can also handle other AD images, since the three AD images (action-class x^a , starting point x^s , ending point x^e) are all similarly designed to perform classification.

To produce the three AD images, one possible way is to perform our methods once for each AD image, i.e., producing them separately. Another option is to stitch the images together into a *combined image* and perform a combined processing for all three AD images. Specifically, we can concatenate the three AD images $\{x_t^a, x_t^s, x_t^e\}$ horizontally to obtain the combined image as $x_t^{combined} \in \mathbb{R}^{N \times (C + 4)}$, where each row consists of three discrete distributions. At the same time, we can still handle the three discrete distributions separately during the diffusion process. To process the stitched image with the diffusion network, we only need to modify the column dimensionality, i.e., by adding 4 columns to the Row-Column Block design above.

There are two advantages in such combined processing. Firstly, it is more efficient and allows us to produce the three AD images in parallel at one go. Secondly, the combined processing also facilitates more sharing of knowledge between the classification and starting/ending point localization sub-tasks, that can be learned via the combined end-to-end update. For instance, by producing the three AD images simultaneously, our model learns to tackle action classification with the knowledge of the global temporal structure of all action instances in the video (gained from the localization sub-tasks), which leads to better performance.

3.4. Inference and Training Pipeline

Inference Pipeline. After obtaining the outputs $\hat{x}_0^a, \hat{x}_0^s, \hat{x}_0^e$, we use them to obtain the final action instances. We largely follow the post-processing pipeline of [30, 31], as follows: First, we find the frames where actions are likely to start or end, by finding pixels in the left column of \hat{x}_0^s, \hat{x}_0^e that are

above a pre-defined threshold δ . Next, because these pixels are often found in clusters, we group up the pixels that are connected, and identify their average position as the starting or ending point. Then, following previous approaches [30, 31] to generate candidate proposals, each starting point \hat{x}_0^s is coupled with all the ending points behind to identify action candidates, with the action duration ranging between the identified starting and ending points. To obtain the action candidate’s class, we average the rows of \hat{x}_0^a corresponding to the duration of the action candidate, and take the class with the highest value. After obtaining all the action candidates for the video sequence, we utilize Soft-NMS [4] to remove overlapping candidates and produce the final results.

Training Pipeline. We use an off-the-shelf model to extract video features f_{ST} , which is kept frozen throughout. At the start, we randomly initialize the diffusion model d with the architecture in Sec. 3.3. During training, we obtain supervision signals for the intermediate steps via the forward process (Eq. 4). Then, we perform the reverse process (Eq. 6) with our diffusion model d to obtain AD image predictions for the intermediate steps and output. We apply the MSE loss between our AD image predictions and the supervision signals at each step, to update the parameters of the diffusion model d .

4. Experiments

4.1. Implementation Details

Following existing works [51, 69], we use an off-the-shelf I3D [7] and R(2+1)D [59] models to extract video features f_{ST} for THUMOS14 and ActivityNet-1.3 respectively. The diffusion network d_ϕ is randomly initialized following the Xavier initialization scheme [18]. Following previous image diffusion works [22, 54], we adopt MSE loss for training. We use AdamW [38] as the optimizer. The initial learning rate is set to 2×10^{-5} and decays following cosine rule. The training batch size is set to 16. Following previous work [69], all training videos are padded to be 2,304 frames and mask operations are added accordingly for redundant padded frames. We set the hyperparameters $T = 50$, $L = 3$, $\delta = 0.9$, $M = 10$. $K = 200$ for THUMOS14 and $K = 2000$ for ActivityNet-1.3. All experiments are conducted on Nvidia V100 GPUs. Refer to Supp for more implementation details.

4.2. Datasets and Evaluation Metric

Following previous works [31, 40, 42, 50, 51, 62, 65, 69], we evaluate our method on the THUMOS14 and ActivityNet-1.3 datasets. **THUMOS14** [23] includes 413 untrimmed videos containing 20 classes of actions. The THUMOS14 dataset is split into a validation set with 200 videos and a test set with 213 videos. We follow existing settings [3, 29, 61, 69] to train our model on the validation set and test on the test set. **ActivityNet-1.3** [6] contains over 20K videos and 200

Table 1. Results on THUMOS14 and ActivityNet-1.3 datasets.

Methods	Feature	THUMOS14					ActivityNet-1.3					
		0.3	0.4	0.5	0.6	0.7	Avg	0.5	0.75	0.95	Avg	
BMN [31]	TSN	56.0	47.4	38.8	29.7	20.5	38.5	TSN	50.1	34.8	8.3	33.9
DBG [28]	TSN	57.8	49.4	39.8	30.2	21.7	39.8	-	-	-	-	
G-TAD [65]	TSN	54.5	47.6	40.3	30.8	23.4	39.3	TSN	50.4	34.6	9.0	34.1
BC-GNN [3]	TSN	57.1	49.1	40.4	31.2	23.1	40.2	TSN	50.6	34.8	9.4	34.3
TAL-MR [71]	I3D	53.9	50.7	45.4	38.0	28.5	43.3	I3D	43.5	33.9	9.2	30.2
P-GCN [68]	R(2+1)D	69.1	63.3	53.5	40.5	26.0	50.5	I3D	48.3	33.2	3.3	31.1
TSA-Net [19]	P3D	61.2	55.9	46.9	36.1	25.2	45.1	P3D	48.7	32.0	9.0	31.9
MUSES [34]	I3D	68.9	64.0	56.9	46.3	31.0	-	I3D	50.0	35.0	6.6	34.0
TCANet [45]	TSN	60.6	53.2	44.6	36.8	26.7	44.3	TSN	52.3	36.7	6.9	35.5
BMN-CSA [57]	TSN	64.4	58.0	49.2	38.2	27.8	47.7	TSN	52.4	36.2	5.2	35.4
ContextLoc [73]	I3D	68.3	63.8	54.3	41.8	26.2	50.9	I3D	56.0	35.2	3.6	34.2
VSGN [70]	TSN	66.7	60.4	52.4	41.0	30.4	50.2	R(2+1)D	53.3	36.8	8.1	35.9
RTD-Net [58]	I3D	68.3	62.3	51.9	38.8	23.7	49.0	I3D	47.2	30.7	8.6	30.8
A ² Net [66]	I3D	58.6	54.1	45.5	32.5	17.2	41.6	I3D	43.6	28.7	3.7	27.8
GTAN [37]	P3D	57.8	47.2	38.8	-	-	-	P3D	52.6	34.1	8.9	34.3
PBRNet [33]	I3D	58.5	54.6	51.3	41.8	29.5	-	I3D	54.0	35.0	9.0	35.0
TaDTR [35]	I3D	62.4	57.4	49.2	37.8	26.3	46.6	I3D	49.1	32.6	8.5	32.3
AFSD [24]	I3D	67.3	62.4	55.5	43.7	31.1	52.0	I3D	52.4	35.3	6.5	34.4
TAGS [40]	I3D	68.6	63.8	57.0	46.3	31.8	52.8	I3D	56.3	36.8	9.6	36.5
STPT [62]	STPT	70.6	65.7	56.4	44.6	30.5	53.6	STPT	51.4	33.7	6.8	33.4
ReAct [50]	3DCNN	69.2	65.0	57.1	47.8	35.6	55.0	3DCNN	49.6	33.0	8.6	32.6
ActionFormer [69]	I3D	82.1	77.8	71.0	59.4	43.9	66.8	R(2+1)D	54.7	37.8	8.4	36.6
DiffTAD [42]	I3D	74.9	72.8	71.2	62.9	58.5	68.0	I3D	56.1	36.9	9.0	36.1
Self-DETR [25]	I3D	74.6	69.5	60.0	47.6	31.8	56.7	I3D	52.2	33.6	8.4	33.7
TriDet [51]	I3D	83.6	80.1	72.9	62.4	47.4	69.3	R(2+1)D	54.7	38.0	8.4	36.8
Ours	I3D	84.9	81.5	76.5	63.0	48.0	70.8	R(2+1)D	56.9	38.9	9.1	38.3

action categories. It comprises of training, validation and test splits containing 10,024, 4,926 and 5,044 videos respectively. Following previous settings [30, 31, 65, 69], our model is optimized on the training set and tested on the validation set.

Evaluation Metric. Following previous works [40, 50, 62, 69], we report the mean average precision (mAP) at different temporal intersection over union (tIoU) thresholds. The tIoU threshold determines how much overlap is required between the prediction and the ground truth to be considered an accurate prediction. We also report the average mAP (Avg), where we average across several tIoUs.

4.3. Main Experimental Results

We compare with state-of-the-art action detection methods on THUMOS14 and ActivityNet-1.3 datasets in Tab. 1. Our proposed method achieves the best results on average mAP among existing methods, showing its efficacy.

4.4. Ablation Studies

Following previous works [50, 51, 62, 69], we conduct ablation experiments on THUMOS14.

Impact of Main Components of ADI-Diff Framework. First, we evaluate the efficacy of our proposed Discrete Action-Detection Diffusion process by comparing against the following baselines: **(A) Stand. Diff. + Model Architecture from [22]:** We apply the standard diffusion process [22] with the image diffusion model architecture from [22]. **(B) Disc. AD Diff. + Model Architecture from [22]:** We adopt our Discrete Action-Detection Diffusion, but use the image diffusion model architecture from [22]. **(C) Stand. Diff. + Row-Col:** We apply the standard diffusion process [22] with our Row-Column Transformer. As observed in Tab. 2, Baseline B which uses the proposed diffusion process obtains a much better result than Baseline A which uses standard diffusion, showing the efficacy of the proposed diffusion process. Notably, this trend also persists when the Row-Column Transformer is used, where our method

significantly outperforms Baseline C. This improvement is because our Discrete Action-Detection Diffusion allows us to effectively map the noisy distributions to the underlying target distribution.

Next, we also validate the efficacy of our proposed Row-Column Transformer design. First, we compare Baseline A vs Baseline C, as well as Baseline B vs our method, and find that the proposed Row-Column Transformer leads to performance improvements in both cases, no matter if the standard diffusion or our proposed diffusion process is used. This shows the efficacy of the proposed Row-Column Transformer. Besides, we further ablate the design of the Row-Column Transformer by comparing against the following alternative designs while applying our proposed diffusion process: **(D) Disc. AD Diff. + Model Architecture from [46]** adopts the model architecture from [46]; **(E) Disc. AD Diff. + Row-Col (w/ Col design only)** adopts an alternative network design (with approximately same network size) that processes both the columns and rows the same way, with MHSA layers only; **(F) Disc. AD Diff. + Row-Col (w/ Row design for both)** adopts an alternative network design (with approximately same network size) that processes both the columns and rows the same way, with TC+MHSA layers; **(G) Disc. AD Diff. + Row-Col (w/o Learnable PE)** replaces the learnable positional embedding with a fixed one generated using the sinusoidal function. Overall, as shown in Tab. 2, our proposed design performs the best, showing its efficacy in capturing both class-wise (across columns) and temporal relationships (across rows).

Table 2. Ablation study for main components of ADI-Diff.

Method	0.3	0.5	0.7	Avg
(A) Stand. Diff. + Model Architecture from [22]	80.4	68.2	44.1	66.0
(B) Disc. AD Diff. + Model Architecture from [22]	82.6	74.3	45.2	69.0
(C) Stand. Diff. + Row-Col	82.1	73.9	45.0	68.1
(D) Disc. AD Diff. + Model Architecture from [46]	82.0	74.1	45.2	67.5
(E) Disc. AD Diff. + Row-Col (w/ Col design for both)	81.8	75.5	45.0	69.0
(F) Disc. AD Diff. + Row-Col (w/ Row design for both)	82.0	75.3	45.1	68.8
(G) Disc. AD Diff. + Row-Col (w/o Learnable PE)	82.4	75.2	45.9	69.1
Ours (Disc. AD Diff. + Row-Col)	84.9	76.5	48.0	70.8

Visualization of Diffusion Process. In Fig. 3, we visualize the action-class AD images produced throughout the reverse diffusion process. We observe that our ADI-Diff framework progressively denoises the original noisy discrete probability distributions, to produce high-quality discrete action-class distributions. See Supp for more results.

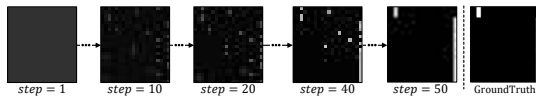


Figure 3. Visualization of diffusion process.

We also compare against a **Standard** baseline, which applies a standard diffusion process [22]. As observed in Fig. 4, our proposed method produces better predictions qualitatively as compared to the Standard baseline. Specifically, the baseline’s action-class AD image (right of Fig. 4) shows much ambiguity and confusion, where the class predictions can be spread over multiple columns (i.e., white pixels

are not concentrated in a consistent column). In contrast, the action-class AD image produced by our method (left of Fig. 4) tends to consistently provide the correct action-class – here, the predictions are concentrated on two separate columns, indicating that two action-classes are captured in this video clip, with one action happening after the other.

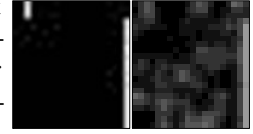


Figure 4. Comparison between the action-class AD image generated by our method (left) and standard diffusion (right).

Impact of Image Stitching. Next, we explore the impact of stitching three of our AD images into a combined image for processing. Results are shown in Tab. 3. We find that stitching the images leads to an efficiency gain and some accuracy gains, as it allows us to produce the AD images in parallel while also facilitating the sharing of knowledge.

Table 3. Ablation study for stitching AD images.

Setting	0.3	0.5	0.7	Avg	Speed (seconds per clip)
w/o stitching	84.7	76.0	47.8	70.3	0.158
w/ stitching	84.9	76.5	48.0	70.8	0.113

Impact of Temporal Boundary AD Images x^s, x^e . We also investigate the impact of introducing x^s, x^e by evaluating the performance without them, where here we follow previous approaches [51, 69] to directly regress the starting and ending points. In Tab. 4, we observe that the performance is much better when we add x^s, x^e , which shows the importance of introducing the temporal boundary AD images to handle the action detection task.

Table 4. Ablation study for temporal boundary AD images.

Setting	0.3	0.5	0.7	Avg
w/o temporal boundary AD images	80.8	71.5	43.9	67.3
w/ temporal boundary AD images	84.9	76.5	48.0	70.8

Inference Speed. In Tab. 5, we compare our method’s speed against existing methods in terms of seconds per video clip. Our method achieves comparable speed to the state-of-the-art [51], yet significantly outperforms it.

Table 5. Inference speed.

Setting	0.3	0.5	0.7	Avg	Speed (seconds per clip)
DiffTAD [42]	74.9	71.2	58.5	68.0	0.397
TriDet [51]	83.6	72.9	47.4	69.3	0.110
Ours	84.9	76.5	48.0	70.8	0.113

5. Conclusion

In this paper, we tackle action detection by casting it as an image generation problem, and propose an AD Image Diffusion (ADI-Diff) framework to generate target AD images via diffusion. With a Discrete Action-Detection Diffusion Process and a Row-Column Transformer design, we attain state-of-the-art performance on two widely-used datasets.

Acknowledgements. This project is supported by the Ministry of Education, Singapore, under the AcRF Tier 2 Projects (MOE-T2EP20222-0009 and MOE-T2EP20123-0014), National Research Foundation Singapore under its AI Singapore Programme (AISG-100E-2023-121).

References

- [1] CV Amrutha, C Jyotsna, and J Amudha. Deep learning approach for suspicious activity detection from surveillance video. In *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, pages 335–339. IEEE, 2020. [1](#)
- [2] 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](#)
- [3] Yueran Bai, Yingying Wang, Yunhai Tong, Yang Yang, Qiyue Liu, and Junhui Liu. Boundary content graph neural network for temporal action proposal generation. In *European Conference on Computer Vision*, pages 121–137. Springer, 2020. [2](#), [7](#)
- [4] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms—improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569, 2017. [7](#)
- [5] Shyamal Buch, Victor Escorcia, Chuanqi Shen, Bernard Ghanem, and Juan Carlos Niebles. Sst: Single-stream temporal action proposals. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2911–2920, 2017. [2](#)
- [6] 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 conference on computer vision and pattern recognition*, pages 961–970, 2015. [7](#)
- [7] 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. [7](#)
- [8] Shuning Chang, Pichao Wang, Fan Wang, Hao Li, and Jishi Feng. Augmented transformer with adaptive graph for temporal action proposal generation. *arXiv preprint arXiv:2103.16024*, 2021. [2](#)
- [9] 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 *proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1130–1139, 2018. [2](#)
- [10] Guo Chen, Yin-Dong Zheng, Limin Wang, and Tong Lu. Dcan: Improving temporal action detection via dual context aggregation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 248–257, 2022. [1](#), [2](#)
- [11] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19830–19843, 2023. [3](#)
- [12] Rui Dai, Srijan Das, Kumara Kahatapitiya, Michael S Ryoo, and François Brémond. Ms-tct: multi-scale temporal conv-transformer for action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20041–20051, 2022. [6](#)
- [13] Stéphane d’Ascoli, Hugo Touvron, Matthew L Leavitt, Ari S Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. In *International Conference on Machine Learning*, pages 2286–2296. PMLR, 2021. [6](#)
- [14] Lin Geng Foo, Jia Gong, Hossein Rahmani, and Jun Liu. Distribution-aligned diffusion for human mesh recovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9221–9232, 2023. [2](#)
- [15] Lin Geng Foo, Hossein Rahmani, and Jun Liu. Ai-generated content (aigc) for various data modalities: A survey. *arXiv preprint arXiv:2308.14177*, 2, 2023. [2](#)
- [16] Jiyang Gao, Zhenheng Yang, Kan Chen, Chen Sun, and Ram Nevatia. Turn tap: Temporal unit regression network for temporal action proposals. In *Proceedings of the IEEE international conference on computer vision*, pages 3628–3636, 2017. [1](#), [2](#)
- [17] Silvio Giancola, Mohieddine Amine, Tarek Dghaily, and Bernard Ghanem. SoccerNet: A scalable dataset for action spotting in soccer videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 1711–1721, 2018. [1](#)
- [18] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010. [7](#)
- [19] Guoqiang Gong, Liangfeng Zheng, and Yadong Mu. Scale matters: Temporal scale aggregation network for precise action localization in untrimmed videos. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2020. [7](#)
- [20] Jia Gong, Lin Geng Foo, Zhipeng Fan, Qihong Ke, Hossein Rahmani, and Jun Liu. Diffpose: Toward more reliable 3d pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. [2](#), [3](#)
- [21] Hongji Guo, Zhou Ren, Yi Wu, Gang Hua, and Qiang Ji. Uncertainty-based spatial-temporal attention for online action detection. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV*, pages 69–86. Springer, 2022. [1](#)
- [22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#)
- [23] 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. [7](#)
- [24] Haohao Jiang, Yao Lu, and Jing Xue. Automatic soccer video event detection based on a deep neural network combined cnn and rnn. In *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 490–494. IEEE, 2016. [1](#)
- [25] Jihwan Kim, Miso Lee, and Jae-Pil Heo. Self-feedback detr for temporal action detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10286–10296, 2023. [7](#)

- [26] Jin Li, Xianglong Liu, Zhuofan Zong, Wanru Zhao, Mingyuan Zhang, and Jingkuan Song. Graph attention based proposal 3d convnets for action detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4626–4633, 2020. [1](#), [2](#)
- [27] 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](#)
- [28] Chuming Lin, Jian Li, Yabiao Wang, Ying Tai, Donghao Luo, Zhipeng Cui, Chengjie Wang, Jilin Li, Feiyue Huang, and Rongrong Ji. Fast learning of temporal action proposal via dense boundary generator. In *Proceedings of the AAAI conference on artificial intelligence*, pages 11499–11506, 2020. [7](#)
- [29] Chuming Lin, Chengming Xu, Donghao Luo, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. Learning salient boundary feature for anchor-free temporal action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3320–3329, 2021. [1](#), [2](#), [5](#), [7](#)
- [30] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. [1](#), [2](#), [6](#), [7](#)
- [31] 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](#), [6](#), [7](#)
- [32] Daochang Liu, Qiyue Li, Anh-Dung Dinh, Tingting Jiang, Mubarak Shah, and Chang Xu. Diffusion action segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10139–10149, 2023. [2](#)
- [33] Qinying Liu and Zilei Wang. Progressive boundary refinement network for temporal action detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11612–11619, 2020. [2](#), [7](#)
- [34] 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](#)
- [35] Xiaolong Liu, Qimeng Wang, Yao Hu, Xu Tang, Shiwei Zhang, Song Bai, and Xiang Bai. End-to-end temporal action detection with transformer. *IEEE Transactions on Image Processing*, 31:5427–5441, 2022. [1](#), [2](#), [7](#)
- [36] Yuan Liu, Lin Ma, Yifeng Zhang, Wei Liu, and Shih-Fu Chang. Multi-granularity generator for temporal action proposal. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3604–3613, 2019. [1](#), [2](#)
- [37] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei. Gaussian temporal awareness networks for action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 344–353, 2019. [7](#)
- [38] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. [7](#)
- [39] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2837–2845, 2021. [2](#)
- [40] Sauradip Nag, Xiatian Zhu, Yi-Zhe Song, and Tao Xiang. Proposal-free temporal action detection via global segmentation mask learning. In *European Conference on Computer Vision*, pages 645–662. Springer, 2022. [2](#), [5](#), [7](#)
- [41] Sauradip Nag, Xiatian Zhu, Yi-Zhe Song, and Tao Xiang. Semi-supervised temporal action detection with proposal-free masking. In *European Conference on Computer Vision*, pages 663–680. Springer, 2022. [2](#)
- [42] Sauradip Nag, Xiatian Zhu, Jiankang Deng, Yi-Zhe Song, and Tao Xiang. Diffvad: Temporal action detection with proposal denoising diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10362–10374, 2023. [2](#), [3](#), [7](#), [8](#)
- [43] Megha Nawhal and Greg Mori. Activity graph transformer for temporal action localization. *arXiv preprint arXiv:2101.08540*, 2021. [2](#)
- [44] Henry Friday Nweke, Ying Wah Teh, Ghulam Mujtaba, and Mohammed Ali Al-Garadi. Data fusion and multiple classifier systems for human activity detection and health monitoring: Review and open research directions. *Information Fusion*, 46: 147–170, 2019. [1](#)
- [45] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 485–494, 2021. [7](#)
- [46] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. [8](#)
- [47] Supriya Sathyanarayana, Ravi Kumar Satzoda, Suchitra Sathyanarayana, and Srikanthan Thambipillai. Vision-based patient monitoring: a comprehensive review of algorithms and technologies. *Journal of Ambient Intelligence and Humanized Computing*, 9:225–251, 2018. [1](#)
- [48] Muhammad Bilal Shaikh and Douglas Chai. Rgb-d data-based action recognition: A review. *Sensors*, 21(12):4246, 2021. [1](#)
- [49] Wenkang Shan, Zhenhua Liu, Xinfeng Zhang, Zhao Wang, Kai Han, Shanshe Wang, Siwei Ma, and Wen Gao. Diffusion-based 3d human pose estimation with multi-hypothesis aggregation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14761–14771, 2023. [2](#)
- [50] 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*, pages 105–121. Springer, 2022. [5](#), [7](#)
- [51] Dingfeng Shi, Yujie Zhong, Qiong Cao, Lin Ma, Jia Li, and Dacheng Tao. Tridet: Temporal action detection with relative

- boundary modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18857–18866, 2023. [5](#), [7](#), [8](#)
- [52] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data. In *The Eleventh International Conference on Learning Representations*, 2023. [2](#)
- [53] 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. [2](#)
- [54] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. [2](#), [3](#), [4](#), [5](#), [6](#), [7](#)
- [55] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019. [1](#), [2](#), [4](#)
- [56] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020. [1](#), [4](#)
- [57] 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. [7](#)
- [58] Jing Tan, Jiaqi Tang, Limin Wang, and Gangshan Wu. Relaxed transformer decoders for direct action proposal generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13526–13535, 2021. [2](#), [7](#)
- [59] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. [7](#)
- [60] Elahe Vahdani and Yingli Tian. Deep learning-based action detection in untrimmed videos: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. [1](#), [5](#)
- [61] Limin Wang, Yu Qiao, Xiaoou Tang, and Luc Van Gool. Actionness estimation using hybrid fully convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2708–2717, 2016. [2](#), [7](#)
- [62] Yuetian Weng, Zizheng Pan, Mingfei Han, Xiaojun Chang, and Bohan Zhuang. An efficient spatio-temporal pyramid transformer for action detection. In *European Conference on Computer Vision*, pages 358–375. Springer, 2022. [1](#), [2](#), [7](#)
- [63] Huijuan Xu, Abir Das, and Kate Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *Proceedings of the IEEE international conference on computer vision*, pages 5783–5792, 2017. [1](#), [2](#)
- [64] Li Xu, Haoxuan Qu, Yujun Cai, and Jun Liu. 6d-diff: A keypoint diffusion framework for 6d object pose estimation. *arXiv preprint arXiv:2401.00029*, 2023. [2](#)
- [65] 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*, pages 10156–10165, 2020. [1](#), [2](#), [7](#)
- [66] Le Yang, Houwen Peng, Dingwen Zhang, Jianlong Fu, and Junwei Han. Revisiting anchor mechanisms for temporal action localization. *IEEE Transactions on Image Processing*, 29:8535–8548, 2020. [7](#)
- [67] Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2678–2687, 2016. [2](#)
- [68] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7094–7103, 2019. [2](#), [7](#)
- [69] 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. [1](#), [2](#), [5](#), [7](#), [8](#)
- [70] 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. [1](#), [2](#), [7](#)
- [71] Peisen Zhao, Lingxi Xie, Chen Ju, Ya Zhang, Yanfeng Wang, and Qi Tian. Bottom-up temporal action localization with mutual regularization. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020. Proceedings, Part VIII 16*, pages 539–555. Springer, 2020. [7](#)
- [72] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2914–2923, 2017. [2](#)
- [73] 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. [1](#), [7](#)