Masked Diffusion Vision-Language Models for Temporal Action Localization

Fengshun Wang[†]
Wuhan University
ycwfs6@whu.edu.cn

Zhengbo Zhang[†]
Singapore University of Technology and Design
zhengbo_zhang@mymail.sutd.edu.sg

Zhigang Tu[*]
Wuhan University
tuzhigang@whu.edu.cn

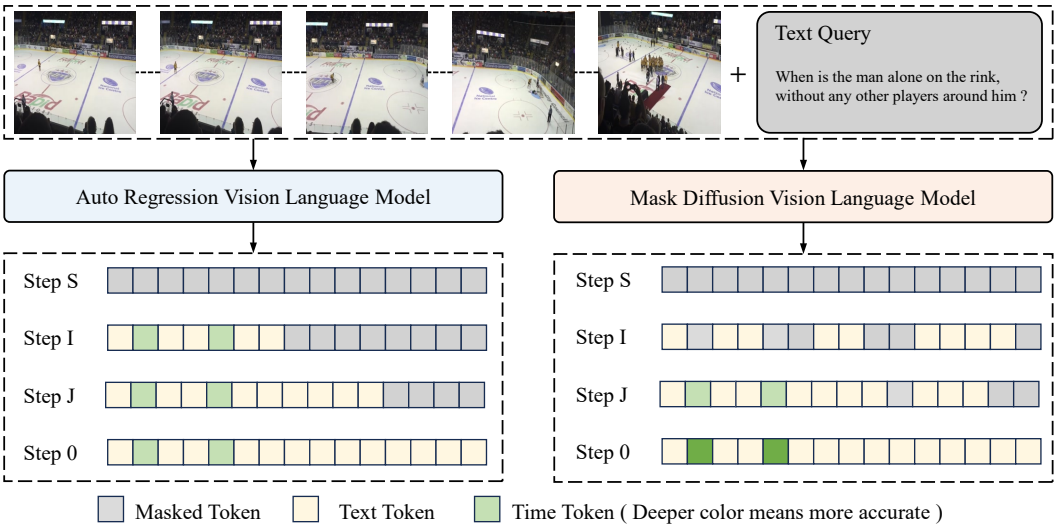


Figure 1: Autoregressive vision-language approaches generate tokens from left to right, limiting their ability to refine earlier predictions using future context. In contrast, masked diffusion vision-language models attend to the full sequence at every step with bidirectional attention, allowing joint refinement of temporal boundaries and semantic descriptions.

Abstract

Temporal action localization (TAL) requires recognizing the target event and localizing its start and end times precisely in untrimmed videos. Recent vision-language formulations improve semantic reasoning and support language-conditioned outputs, but their autoregressive decoders still generate tokens from left to right, preventing later semantic evidence from revising earlier timestamp predictions. We adapt masked diffusion vision-language models (MDVLMs) to TAL so that semantic tokens and boundary tokens remain editable throughout iterative denoising with bidirectional attention, allowing temporal boundaries and semantic content to be refined jointly. Direct adaptation, however, creates two TAL-specific mismatches: standard masked diffusion training corrupts all positions uniformly at random, but the time tokens are more reliable when enough semantic context is available; and token-level cross-entropy does not reflect temporal IoU. To address these mismatches, we introduce a **Planned Training Objective** that uses boundary-aware masking and step-weighted reconstruction to rehearse the late recovery of time

[†]Equal contribution.
[*]Corresponding author: tuzhigang@whu.edu.cn

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

tokens, together with a **Step-Level IoU Reward** that provides overlap-aware supervision during denoising. A standard sequence-level cross-entropy term provides the base reconstruction signal. Experiments on ActivityNet-RTL, ActivityNet-1.3, and THUMOS-14 show that MDVLM-TAL improves both temporal reasoning and boundary localization over autoregressive vision-language baselines, with especially strong gains under stricter temporal IoU criteria. Our source code is available at `https://github.com/ycwfs/MDVLM-TAL`.

1 Introduction

Temporal Action Localization (TAL) requires localizing the temporal extent of each action instance in an untrimmed video. A correct prediction couples two decisions, recognizing the queried event category and placing its start and end times precisely. Because boundary errors of even a few seconds push the temporal IoU below standard evaluation thresholds, TAL demands stricter temporal precision than video-level classification or captioning.

Classical TAL pipelines treat localization as visual proposal classification and regression [2, 18, 43, 52]. They extract temporal features, generate candidate segments or regress boundary offsets, and classify each segment with action-specific heads. These detectors excel on closed-set benchmarks. However, they are not well-suited for settings that require language conditioned undertanding, detailed event semantics. To deal with the problems above, some researchers have innovated using the powerful contextual understanding ability of Large Vision-Language Models (LVLM) by treating TAL as generative task with LVLM [5, 8, 9, 14, 19, 42, 45, 48, 49, 53] (see Figure 1). The video and query are encoded jointly, timestamps are discretized into special time tokens, and the action label, boundary tokens, and optional text are generated as one sequence in [9]. This formulation grants TAL access to language priors and free form explanations. However, LVLM-based TAL systems still inherit the autoregressive (AR) decoder of their backbone, leaving a gap between the context understanding ability and the temporal action localization ability of the AR model [1, 38–40]. This gap is caused by the left to right token generation order of AR and the immutability of token generation. And it can be optimized through iterative updates.

As shown in Figure 1, the recent language model paradigm, Masked diffusion vision-language models (MDVLMs) [16], can replace this left to right commitment with iterative denoising under bidirectional attention [26]. Generation begins from a fully masked response and progressively unmasks tokens with global context at every step, so semantic tokens and time tokens remain editable until the final step. This is the structural property TAL requires, boundary tokens are revised after more semantic evidence has appeared, and the start, end, and text tokens are refined within the same generation process.

With these in mind, in this paper, we aim to leverage MDVLMs to address TAL. Yet, this is non-trivial, treating TAL as a generic masked diffusion problem leaves two TAL-specific mismatches unaddressed. **(i) Schedule mismatch.** The response is a *mixed* sequence of ordinary text tokens and time tokens that encode boundaries. Our dataset-level statistics show that semantic tokens are revealed earlier than time tokens and that boundary predictions stabilize later in denoising (Figure 4), whereas standard masked diffusion training corrupts all positions uniformly at random and never rehearses this late reveal behavior. **(ii) Loss mismatch.** TAL is evaluated by temporal IoU, which depends on the numerical *distance* between predicted and ground-truth boundaries, but token-level losses penalize all wrong tokens equally and fail to reward an update from $\langle t_0 \rangle$ to $\langle t_{47} \rangle$ when the target is $\langle t_{43} \rangle$.

To tackle these challenges, we propose **MDVLM-TAL**, a masked diffusion vision-language framework that closes both gaps in the training objective. The **Planned Training Objective** uses boundary-aware masking and step-weighted reconstruction so that time tokens are typically recovered later than semantic tokens during training, matching their behavior at inference. The **Step-Level IoU Reward** provides overlap-aware supervision during denoising. A standard sequence-level cross-entropy term provides the base reconstruction loss. Our contributions are:

- We formulate TAL as masked diffusion sequence generation, so semantic tokens and boundary time tokens are refined jointly through bidirectional denoising rather than fixed by a left to right order.
- We introduce a **Planned Training Objective** that uses boundary-aware masking and step-weighted reconstruction to delay the recovery of time tokens relative to semantic tokens.

- We propose a **Step-Level IoU Reward** that injects overlap-aware supervision into intermediate denoising states for more precise temporal localization.
- On THUMOS-14, ActivityNet-1.3, and ActivityNet-RTL, MDVLM-TAL surpasses both autoregressive vision-language baselines and the strongest specialized TAL detectors, with the largest gains at strict tIoU thresholds.

2 Related Work

Temporal Action Localization. Temporal action localization has evolved from proposal-based and two-stage pipelines, such as TAL-Net [2], BMN [18], G-TAD [43], RTD-Net [35], TCANet [29], and ContextLoc [62], to stronger one-stage and transformer-centric detectors, including ReAct [32], ActionFormer [52], TALLFormer [3], TadTR [21], TriDet [33], TemporalMaxer [36], TFFormer [47], TransGMC [46], RefineTAD [4], DualDETR [61], Pred-DETR [11], and CLTDR [15]. Recent work further expands TAL beyond the standard fully supervised closed-set setting, studying weakly supervised localization [10, 13, 30, 51, 60], zero-shot or open-vocabulary detection [17, 23, 28, 37, 41], online localization [31, 34], and generative or denoising formulations such as DiffTAD and DenoiseLoc [25, 44]. OpenTAD [20] further shows that evaluation conclusions can change substantially once methods are reimplemented under a unified framework. Still, most TAL methods remain detector-style models that directly classify and regress intervals from visual features, rather than generating temporally grounded multimodal outputs.

Vision-Language Models for Temporal Action Localization. Recent generative vision-language methods represent boundaries as timestamp tokens and cast localization as text generation, including LITA, VTimeLLM, VTG-LLM, and UniMD [5, 8, 9, 49, 57]. A related line uses MLLMs as auxiliary supervision for weakly supervised TAL rather than as the decoder itself [54]. These methods show that vision-language modeling is useful for temporal localization, but they mostly rely on autoregressive vision-language decoding or use the vision-language model as an external guide. Our work is closest to the generative line, but replaces left to right decoding with masked diffusion. but replaces left to right decoding with masked diffusion. This design allows all timestamp tokens to be predicted jointly rather than sequentially, mitigating error accumulation in boundary generation. It also enables iterative refinement of temporal boundaries within a single denoising process.

Diffusion Models for Perception and Structured Generation. Text-to-image diffusion models also provide useful representations for visual perception. VPD [59] adapts their pretrained denoising features and cross-attention maps to perception tasks, while TADP [12] improves these representations through caption-based text–image alignment. Related work applies text-to-image diffusion models to category-agnostic pose estimation [27] and unsupervised visual tracking, where Diff-Tracker learns and updates target-specific prompts [55]. These studies [56, 58] demonstrate the value of generative semantic knowledge for spatial prediction. MDVLM-TAL addresses temporal localization by jointly denoising both token types within a language response, with a task-specific recovery schedule and temporal-overlap supervision.

3 Method

3.1 Preliminaries

Task definition. Given an untrimmed video $V = \{f_n\}_{n=1}^{T_v}$ and an instruction q, our goal is to generate a response that contains temporal boundaries and optional explanation text. We consider two regimes. In reasoning temporal localization (e.g., ActivityNet-RTL), q specifies one target event and the response contains one boundary pair plus text. In standard closed-set TAL benchmarks, q is class-agnostic and the model must infer all action instances directly from video content.

Time-token formulation. We discretize the timeline into N bins and represent boundaries with special tokens from $\mathcal{V}_{\text{time}} = \{\langle t_0\rangle, \ldots, \langle t_{N-1}\rangle\}$. Let $\mathcal{V}$ denote the full output vocabulary, let $Y^\star = [y_1, \ldots, y_L]$ denote the target response sequence, and let $\hat{B} = [\hat{b}_s, \hat{b}_e]$ be the segment decoded from the generated time tokens. If the ground-truth boundary indices are $(s^\star, e^\star)$, the normalized timestamps are $b_s^\star = \frac{s^\star}{N-1}, b_e^\star = \frac{e^\star}{N-1}, 0 \le b_s^\star \le b_e^\star \le 1$. Localization quality is measured by the temporal IoU between the predicted segment $\hat{B}$ and the ground-truth segment $B^\star = [b_s^\star, b_e^\star]$.

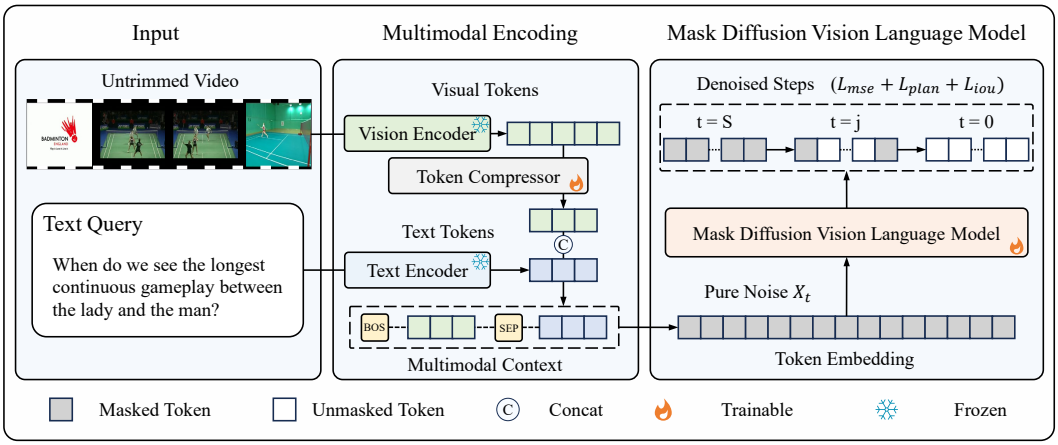


Figure 2: Overview of MDVLM-TAL. Given a video and a query, a vision and text encoder are used to encode the inputs. Then, we compress the vision token with a token compressor. Video and text tokens are concatenated with noisy tokens to build a multimodal context. The masked diffusion backbone jointly denoises semantic tokens and boundary time tokens using global bidirectional attention. The Planned Training Objective keeps time tokens masked longer than semantic tokens during training, and the Step-Level IoU Reward provides overlap-aware supervision during denoising.

Masked Diffusion Large Language Models. Let $x_0 = Y^\star$ be the clean target response. A masked diffusion language model defines a sequence of corrupted states $\{x_t\}_{t=0}^{S}$, where $t = 0$ is the clean sequence and $t = S$ is the fully masked state. For each diffusion step $t \in \{1, \ldots, S\}$, we sample a binary keep mask $m_t \in \{0, 1\}^L$ and construct

$$x_t = m_t \odot x_0 + (1 - m_t) \odot \texttt{[MASK]}. \tag{1}$$

We denote the masked positions by $\mathcal{M}_t = \{i \in \{1, \ldots, L\} : m_{t,i} = 0\}$. Conditioned on the multimodal context C and the current step t, the denoiser parameterized by θ predicts the clean-token distribution

$$p_\theta(x_0 \mid x_t, C, t) = \prod_{i=1}^{L} p_\theta(x_{0,i} \mid x_t, C, t). \tag{2}$$

Training samples a diffusion step and learns to recover the clean tokens at the masked positions. Inference starts from the fully masked state x_S and runs reverse denoising from $t = S$ to $t = 1$, progressively revealing the masked positions with the highest token confidence until the final sequence is obtained. Because every position remains editable until the last steps, masked diffusion is well suited to TAL, where semantic tokens and boundary time tokens should be refined jointly rather than fixed by a left-to-right decoding order.

3.2 Architecture overview.

As shown in the Figure2, we follow the standard *visual encoder + multimodal projector + language backbone* design used in Video-LLM systems [19, 45, 53], we encode the video as $Z_v = \Pi(\mathrm{Enc}_v(V))$, and a token compressor is used to compress and project vision tokens, then we tokenize the instruction as Z_q, and form the multimodal context $C = [\texttt{BOS}, Z_v, \texttt{SEP}, Z_q, \texttt{SEP}]$, where BOS and SEP denote the beginning-of-sequence and separator tokens.

The token compressor consists of one cross-attention block applied independently to each frame. Eight learned query tokens, shared across frames, attend to the spatial tokens of each frame. A feed-forward layer with a residual connection follows the cross-attention operation. This produces eight visual tokens per frame, which we concatenate in frame order to preserve the temporal ordering of the video.

3.3 Planned Training Objective

Standard masked diffusion training uses a uniform random masking process, but in TAL boundary tokens usually become reliable only after enough semantic context has been formed. If timestamps

appear too early during training, the learned corruption path no longer matches the late-reveal behavior of boundary tokens during reverse denoising. Our planned training objective addresses this mismatch by delaying the unmasking of boundary tokens and emphasizing denoising stages where fine temporal cues matter most.

Planned training has two parts: a boundary-aware masking policy and a step-weighted reconstruction loss. For a sampled denoising step t, the masking policy keeps temporal boundary positions masked more aggressively than non-temporal-boundary positions, and the reconstruction loss places larger weight on lower-noise steps that carry finer temporal boundary information.

Boundary-aware masking. We partition token positions into a temporal boundary set $\mathcal{B} = \{i \in \{1, \dots, L\} : y_i^\star \in \mathcal{V}_{\text{time}}\}$, whose target token is a time token, and its complement $\mathcal{N} = \{1, \dots, L\} \backslash \mathcal{B}$. For step t, we define type-specific keep probabilities as $\kappa_t^{\mathcal{B}} = \left(1 - \frac{t}{S}\right)^\gamma, \kappa_t^{\mathcal{N}} = \left(1 - \frac{t}{S}\right)^\eta, \gamma > \eta > 0$, so $\kappa_t^{\mathcal{B}} < \kappa_t^{\mathcal{N}}$ for $t \in (0, S)$ and boundary tokens remain masked longer than non-boundary tokens. We then sample the training mask as

$$m_{t,i} \sim \begin{cases} \text{Bernoulli}(\kappa_t^{\mathcal{B}}), & i \in \mathcal{B}, \\ \text{Bernoulli}(\kappa_t^{\mathcal{N}}), & i \in \mathcal{N}. \end{cases} \tag{3}$$

This induces a boundary-aware corruption distribution $q_\phi(x_t \mid x_0, t) = \prod_{i=1}^{L} q_\phi(x_{t,i} \mid x_{0,i}, t)$, where $\phi = \{\gamma, \eta\}$ parameterizes the planned masking policy and x_t is formed from m_t as in the preliminaries.

Step-weighted reconstruction. We train the denoiser under the planned corruption process with cross-entropy restricted to the masked positions:

$$\mathcal{L}_{\text{plan}} = \mathbb{E}_{(V,q,Y^\star)\sim\mathcal{D}} \mathbb{E}_{t\sim\mathcal{U}\{1,\dots,S\}} \mathbb{E}_{x_t\sim q_\phi(\cdot|x_0,t)} \left[w_t \sum_{i:\, m_{t,i}=0} -\log p_\theta(x_{0,i} \mid x_t, C, t) \right], \tag{4}$$

where $\sum_{t=1}^{S} w_t = 1$ and $w_{t_1} > w_{t_2}$ if $t_1 < t_2$, so lower-noise denoising stages receive larger weight. Compared with vanilla diffusion training, $\mathcal{L}_{\text{plan}}$ makes the model rehearse the late recovery of time tokens observed during reverse denoising, instead of treating time tokens and semantic tokens as equally easy to reveal at every step.

3.4 Step-Level IoU Reward

Cross-entropy supervises token identity rather than temporal distance. In TAL, predicting $\langle t_{47} \rangle$ is much better than predicting $\langle t_0 \rangle$ when ground truth is $\langle t_{43} \rangle$, but standard token losses do not capture this difference. We therefore add step-level supervision that measures whether each denoising update improves the current temporal segment.

Our IoU-guided reward has two parts: a soft IoU score derived from the start and end token distributions at each denoising step, and a gating mechanism that activates the reward only after both boundary tokens have been revealed.

Soft boundary reward. Let i_s and i_e denote the start and end slots of the current action tuple in the response template. From the full-vocabulary logits $z_t^{(i_s)}$ and $z_t^{(i_e)}$, we restrict to the time-token vocabulary $\{\langle t_k \rangle\}_{k=0}^{N-1}$ and renormalize:

$$p_t^s(k) = \frac{\exp z_t^{(i_s)}[\langle t_k \rangle]}{\sum_{k'=0}^{N-1} \exp z_t^{(i_s)}[\langle t_{k'} \rangle]}, \qquad p_t^e(k) = \frac{\exp z_t^{(i_e)}[\langle t_k \rangle]}{\sum_{k'=0}^{N-1} \exp z_t^{(i_e)}[\langle t_{k'} \rangle]}. \tag{5}$$

We then compute ordered soft boundaries by expectation as $\tilde{s}_t = \sum_{k=0}^{N-1} k\, p_t^s(k)$, $\tilde{e}_t = \sum_{k=0}^{N-1} k\, p_t^e(k)$, $\bar{b}_{s,t} = \frac{\tilde{s}_t}{N-1}$, and $\bar{b}_{e,t} = \frac{\tilde{e}_t}{N-1}$. We keep the start and end roles distinct and define the soft overlap terms as

$$\ell_t^{\text{pred}} = \max\left(0, \bar{b}_{e,t} - \bar{b}_{s,t}\right), \ell_t^{\cap} = \max\left(0, \min\left(\bar{b}_{e,t}, b_e^\star\right) - \max\left(\bar{b}_{s,t}, b_s^\star\right)\right), \\ \ell_t^{\cup} = \ell_t^{\text{pred}} + (b_e^\star - b_s^\star) - \ell_t^{\cap}, \tag{6}$$

and compute the soft IoU reward as $r_t^{\text{soft}} = \frac{\ell_t^{\cap}}{\ell_t^{\cup} + \epsilon}$.

Gated relative weighting. Let $\hat{y}_t^{(s)}$ and $\hat{y}_t^{(e)}$ denote the currently decoded tokens at the start and end positions after reverse-denoising step t. To avoid noisy supervision from incomplete boundaries, we activate IoU reward only when both boundary tokens are generated at step t. Let $g_t \in \{0, 1\}$ be $g_t = \mathbb{I}\Big[\hat{y}_t^{(s)} \neq \texttt{[MASK]} \ \wedge \ \hat{y}_t^{(e)} \neq \texttt{[MASK]}\Big]$. Next, we compute the group-relative reward statistics over the active denoising steps:

$$\bar{r} = \frac{\sum_{t=1}^{S} g_t r_t^{\text{soft}}}{\epsilon + \sum_{t=1}^{S} g_t}, \qquad \sigma_r = \sqrt{\frac{\sum_{t=1}^{S} g_t \left(r_t^{\text{soft}} - \bar{r}\right)^2}{\epsilon + \sum_{t=1}^{S} g_t} + \epsilon}, \tag{7}$$

and define the normalized step advantage $A_t = g_t \frac{r_t^{\text{soft}} - \bar{r}}{\sigma_r}$. To keep the relative weighting strictly non-negative, we use $\tilde{A}_t = \max(A_t, 0)$. We optimize a unified IoU reward objective:

$$\begin{aligned}\mathcal{L}_{\text{iou}} = \frac{1}{\epsilon + \sum_{t=1}^{S} g_t} \Bigg[& \sum_{t=1}^{S} \alpha_t g_t (1 - r_t^{\text{soft}}) + \mu \sum_{t=2}^{S} g_t g_{t-1} \max\big(0,\, r_t^{\text{soft}} - r_{t-1}^{\text{soft}} + \delta\big) \\ & - \lambda_{\text{rel}} \sum_{t=1}^{S} \text{stopgrad}(\tilde{A}_t) \log\big(p_t^s(s^\star)\, p_t^e(e^\star)\big) \Bigg],\end{aligned} \tag{8}$$

Here α_t are step weights, μ controls the monotonic refinement penalty, δ is the refinement margin, and λ_{rel} controls the relative weighting term. The first term favors high step-wise soft tIoU at every active step. The second encourages soft IoU to improve along the reverse trajectory (from x_t to x_{t-1}, i.e., toward the clean state at $t = 0$): the hinge $\max(0, r_t^{\text{soft}} - r_{t-1}^{\text{soft}} + \delta)$ is non-zero whenever $r_{t-1}^{\text{soft}} < r_t^{\text{soft}} + \delta$, penalizing pairs where overlap fails to improve by at least δ. The third reinforces active steps whose IoU is above the active-step mean without pushing weak steps away from the target. Here $\text{stopgrad}(\cdot)$ means that the normalized weights are detached before they are applied to the boundary log-likelihood.

3.5 Final objective.

Besides the planned training term, the implementation keeps the standard token-level reconstruction loss returned by the backbone. Let $\bar{Y}_t^\star$ denote the supervised label sequence after ignoring positions outside the current training target. We write the base reconstruction loss as $\mathcal{L}_{\text{ce}} = \text{CE}\big(p_\theta(\cdot \mid x_t, C, t), \bar{Y}_t^\star\big)$. And the full training objective is $\mathcal{L} = \lambda_{\text{ce}}\mathcal{L}_{\text{ce}} + \lambda_{\text{plan}}\mathcal{L}_{\text{plan}} + \lambda_{\text{iou}}\mathcal{L}_{\text{iou}}$.

3.6 Inference

Given an input video V and instruction q, we first build the multimodal context C from the encoded video tokens and instruction tokens. Inference starts from the fully masked output sequence x_S and runs reverse denoising for $t = S, S-1, \ldots, 1$. At each step, the denoiser predicts $p_\theta(x_{0,i} \mid x_t, C, t)$ for every masked position. We then reveal the scheduled number of masked positions with highest maximum token probability and keep the remaining positions masked to form x_{t-1}. This process usually resolves semantic tokens earlier and time tokens later, which matches the planned training design. After the final step, the decoded x_0 is parsed into the output response, including the predicted boundary tokens and any associated text.

4 Experiments

4.1 Implementation Details

Datasets and Benchmarks.

We evaluate MDVLM-TAL on three TAL benchmarks: ActivityNet-1.3, THUMOS-14, and ActivityNet-RTL. ActivityNet-1.3 is a large-scale benchmark built on the ActivityNet video collection [6] and emphasizes long untrimmed videos with diverse action classes. THUMOS-14 contains short, densely annotated sports actions and stresses boundary precision under frequent action transitions. ActivityNet-RTL [9] focuses on reasoning and evaluates whether models can jointly produce semantic rationales and accurate temporal boundaries.

Table 1: Comparison with state-of-the-art methods on THUMOS-14 and ActivityNet-1.3.

Method	Stage	THUMOS-14						ActivityNet-1.3			
		0.3	0.4	0.5	0.6	0.7	Avg	0.5	0.75	0.95	Avg
Discriminative models											
TAL-Net [2]	2-stage	53.2	48.5	42.8	33.8	20.8	–	38.2	18.3	1.3	20.2
BMN [18]	2-stage	56.0	47.4	38.8	29.7	20.5	38.5	50.1	34.8	8.3	33.9
G-TAD [43]	2-stage	54.5	47.6	40.3	30.8	23.4	39.3	50.4	34.6	9.0	34.1
RTD-Net [35]	2-stage	68.3	62.3	51.9	38.8	23.7	–	47.2	30.7	8.6	30.8
TCANet [29]	2-stage	60.6	53.2	44.6	36.8	26.7	44.3	52.3	36.7	6.9	35.5
ContextLoc [62]	2-stage	68.3	63.8	54.3	41.8	26.2	50.9	56.0	35.2	3.6	34.2
ReAct [32]	1-stage	69.2	65.0	57.1	47.8	35.6	55.0	49.6	33.0	8.6	32.6
TAGS [24]	1-stage	68.6	63.8	57.0	46.3	31.8	52.8	56.3	36.8	9.6	36.5
TALLFormer [3]	1-stage	76.0	–	63.2	–	34.5	59.2	54.1	36.2	7.9	35.6
ActionFormer [52]	1-stage	82.1	77.8	71.0	59.4	43.9	66.8	54.7	37.8	8.4	36.6
TemporalMaxer [36]	1-stage	82.8	78.9	71.8	60.5	44.7	67.7	–	–	–	–
TFFormer [47]	1-stage	82.1	78.9	72.0	60.8	44.9	67.8	54.4	36.7	7.5	35.8
TransGMC [46]	1-stage	82.3	78.8	71.4	60.0	45.1	67.5	54.8	37.6	8.5	36.7
TriDet [33]	1-stage	83.6	80.1	72.9	62.4	47.4	69.3	54.7	38.0	8.4	36.8
CLTDR-GMG [15]	1-stage	84.1	80.3	73.6	62.4	48.2	69.9	55.0	38.0	8.6	37.1
Generative models											
DiffTAD baseline (10-step) [25]	2-stage	70.0	66.5	60.6	47.5	36.9	56.3	51.0	32.9	9.0	32.4
DiffTAD (5-step) [25]	1-stage	73.4	71.5	69.9	62.8	58.4	67.2	55.2	36.8	8.9	36.0
DiffTAD (10-step) [25]	1-stage	74.9	72.8	71.2	62.9	58.5	68.0	56.1	36.9	9.0	36.1
MDVLM-TAL-8B (ours)	1-stage	**84.6**	**80.8**	**74.4**	**63.5**	**60.1**	**72.7**	**58.2**	**38.8**	**9.8**	**37.9**

Evaluation Metrics. For THUMOS-14 and ActivityNet-1.3 we follow the standard TAL protocol and report per-class mean Average Precision (mAP) at multiple tIoU thresholds, with the official averages over $[0.3{:}0.1{:}0.7]$ and $[0.5{:}0.05{:}0.95]$ respectively. For ActivityNet-RTL we follow the LITA protocol [9] and report mIoU, $P@0.5$, and the GPT-based "Score" for explanation quality under per-video averaging; we additionally report $P@0.75$ and $P@0.95$ for our model and mark them "–" for baselines that did not.

Settings. We use `LLaDA-8B-Instruct` [26] as the masked diffusion language backbone and `SigLIP-SO400M-Patch14-384` [50] as the vision encoder. In our implementation, θ in the the denoiser consists of the LoRA adapters inserted into the language model, together with the multimodal projector and the temporal video compressor when it is enabled. The pretrained backbone weights and the SigLIP vision-tower weights remain frozen by default.

We uniformly sample 100 frames from each video, discretize timestamps into $N = 100$ relative time tokens $\langle t_0\rangle, \ldots, \langle t_{99}\rangle$, use $S = 64$ denoising steps, and set the number of compressor query tokens to 8. For a timestamp τ in a video of duration D, we use $k = \mathrm{round}((N-1)\tau/D)$ during encoding and decode a predicted token by $\tau \approx Dk/(N-1)$. We use LoRA [7] with rank 16, $\alpha = 32$, and dropout 0.05. For the planned masking policy, we fix $(\gamma, \eta) = (2, 1)$, and for the step-weight schedule we use the normalized linear form $w_t = \frac{2(S-t+1)}{S(S+1)}$. For the IoU reward, we use the same step weights $\alpha_t = w_t$, together with $\mu = 0.5$, $\delta = 0.01$, $\lambda_{\text{rel}} = 0.5$, and $\epsilon = 10^{-8}$. Unless otherwise stated, the loss-balance coefficients are $(\lambda_{\text{plan}}, \lambda_{\text{iou}}, \lambda_{\text{ce}}) = (1.0, 0.5, 0.2)$.

For closed-set TAL, the prompt is "Identify all action instances in the video and output each action label with its start and end time stamps." Across closed-set TAL and ActivityNet-RTL, we use AdamW with learning rate 2×10^{-4}, weight decay 0.01, cosine decay, warmup ratio 0.03, per-device batch size 1, and gradient accumulation of 8 steps. In the released scripts, RTL uses 5 epochs with maximum sequence length 4096, while the closed-set event-localization script uses 10 epochs with maximum sequence length 2048. For ablations over loss balance, we report coefficients $(\lambda_{\text{plan}}, \lambda_{\text{iou}}, \lambda_{\text{ce}})$ in Table 4. For RTL inference, we use confidence-based unmask decoding with block length 32 and step ratio 1. All the experiments are conducted on 4 NVIDIA L40S GPUs.

Table 2: Results on ActivityNet-RTL under the LITA evaluation protocol [9]. Qwen3.5-9B is an additional autoregressive (AR) baseline trained on the same data with the same visual encoder, projector, and token compressor. Unreported metrics are marked "–".

Model	Size	mIOU	P@0.5	P@0.75	P@0.95	Score
Video-LLaMA-v2 [53]	13B	–	–	–	–	32.1
Video-ChatGPT [22]	13B	–	–	–	–	38.8
LITA-7B [9]	7B	24.1	21.2	–	–	44.0
LITA-13B [9]	13B	28.6	25.9	–	–	46.3
Qwen3.5-9B (AR)	9B	30.09	26.59	–	–	48.3
MDVLM-TAL (ours)	8B	**31.24**	**27.12**	**21.3**	**6.4**	**48.9**

4.2 Comparison with State-of-the-Art

Table 1 shows that MDVLM-TAL improves the strongest prior results on both closed-set benchmarks. On THUMOS-14, MDVLM-TAL reaches the highest mAP at every reported tIoU threshold, with the largest gains at stricter thresholds such as 0.7. On ActivityNet-1.3, it also improves the official average mAP and the high-overlap metrics. These gains are important because high-threshold mAP measures boundary quality rather than only coarse action recognition. The results support the central claim that temporal boundaries benefit from remaining editable during sequence generation, rather than being produced once by proposal refinement or left to right decoding.

Table 2 evaluates the query-conditioned reasoning setting. ActivityNet-RTL measures localization quality with mIoU and precision-at-threshold, and measures explanation quality with the LITA "Score" [9]. Some general-purpose Video-LLMs in the source benchmark do not reliably emit timestamps, so their comparison is score-only. We keep this reporting convention and mark $P@0.75$ and $P@0.95$ as unavailable when a baseline does not report them. We additionally evaluate Qwen3.5-9B as a stronger autoregressive baseline, using the same SigLIP-SO400M visual encoder, multimodal projector, token compressor, training data, and training settings. MDVLM-TAL achieves the highest reported localization and explanation scores while using a single 8B language backbone.

Table 2 shows that Qwen3.5-9B improves over LITA-13B. MDVLM-TAL further exceeds Qwen3.5-9B by 1.15 percentage points in mIoU, 0.53 points in $P@0.5$, and 0.6 in explanation Score. The advantage persists against this stronger autoregressive baseline under a shared visual frontend and task training setup. Since the language backbones differ, we next isolate the proposed objectives within the same masked diffusion model.

4.3 Ablation Study

Cross-dataset component ablations. Table 3 evaluates all three benchmarks with the same LLaDA-8B backbone, visual frontend, and training data. Base MDVLM retains LoRA adaptation but uses uniform fine-tuning without Planned Training or the Step-Level IoU Reward. The reversed plan recovers time tokens earlier; rewarding every step removes the both-boundary gate. Full fine-tuning replaces LoRA with full-parameter language-backbone updates. Other variants remove one component from the full model.

Contributions of Planned Training. Removing Planned Training lowers average mAP by 1.7 points on THUMOS-14 and 1.0 point on ActivityNet-1.3, and RTL mIoU by 3.03 points. Both planned masking and step weighting contribute, with larger drops from removing masking on all three datasets. Reversing the plan is weaker still, yielding 70.5 and 36.7 average mAP and 27.86 RTL mIoU. The consistent degradation supports recovering semantic context before temporal boundaries across both closed-set detection and query-conditioned reasoning.

Overlap supervision and boundary gating. Removing the Step-Level IoU Reward lowers average mAP by 1.2 and 0.7 points on THUMOS-14 and ActivityNet-1.3, and RTL mIoU by 2.5 points. Rewarding every step yields 30.12 RTL mIoU versus 31.24 for the full model, supporting overlap feedback after both boundaries are available. Boundary-only cross-entropy also underperforms across

Table 3: Component ablations on three benchmarks with the same 8B masked diffusion backbone and training data. "w/o" removes a component from the full model. All metrics are percentages except Score.

Setting	THUMOS-14 (mAP)						ActivityNet-1.3 (mAP)				ActivityNet-RTL		
	0.3	0.4	0.5	0.6	0.7	Avg	0.5	0.75	0.95	Avg	mIoU	P@0.5	Score
MDVLM-TAL (ours)	**84.6**	**80.8**	**74.4**	**63.5**	**60.1**	**72.7**	**58.2**	**38.8**	**9.8**	**37.9**	**31.24**	**27.12**	**48.9**
w/o Planned Training	83.1	79.0	72.8	62.3	57.6	71.0	57.0	37.8	9.3	36.9	28.21	23.86	46.2
w/o planned masking	83.9	79.8	73.4	62.6	58.8	71.7	57.5	38.1	9.4	37.2	29.35	25.14	47.0
w/o step weighting	84.2	80.2	73.8	63.0	59.3	72.1	57.7	38.3	9.5	37.3	30.31	26.02	48.4
w/ reversed plan	82.8	78.7	72.3	61.8	57.0	70.5	56.7	37.5	9.1	36.7	27.86	23.41	45.9
w/o IoU reward	83.8	79.6	73.2	62.4	58.3	71.5	57.4	38.1	9.4	37.2	28.74	24.52	46.5
w/ reward at every step	84.1	80.1	73.7	62.9	59.2	72.0	57.7	38.2	9.5	37.3	30.12	25.86	48.2
w/ boundary-only CE	84.0	79.9	73.5	62.7	58.9	71.8	57.6	38.2	9.4	37.3	29.68	25.43	47.6
Full fine-tuning	84.3	80.3	73.4	63.2	59.4	72.2	57.8	38.5	9.6	37.4	30.58	26.03	47.3
Base MDVLM (w/o both)	82.4	78.2	71.6	61.2	55.9	69.9	56.4	37.2	8.9	36.3	26.94	22.42	45.2

Table 4: Loss-weight ablation on THUMOS-14 and ActivityNet-1.3 for the 8B MDVLM-TAL model.

λ_{plan}	λ_{iou}	λ_{ce}	THUMOS-14					ActivityNet-1.3		
			0.3	0.4	0.5	0.6	0.7	0.5	0.75	0.95
1.0	0.25	0.2	83.9	80.1	73.6	62.8	59.4	57.5	38.1	9.4
1.0	0.5	0.1	84.2	80.4	73.9	63.1	59.7	57.8	38.4	9.6
1.0	**0.5**	**0.2**	**84.6**	**80.8**	**74.4**	**63.5**	**60.1**	**58.2**	**38.8**	**9.8**
1.0	1.0	0.2	84.0	80.2	73.7	62.9	59.5	57.6	38.2	9.5
0.5	0.5	0.2	83.7	79.9	73.4	62.6	59.2	57.3	37.9	9.3

datasets, reaching 29.68 RTL mIoU and 47.6 Score. Full-sequence supervision thus benefits both localization and explanation quality.

Task-specific objectives and adaptation. The full model improves over Base MDVLM by 2.8 and 1.6 points in average mAP on THUMOS-14 and ActivityNet-1.3, and by 4.30 points in RTL mIoU, with LoRA and training data held fixed. Full fine-tuning obtains 72.2 and 37.4 average mAP and 30.58 RTL mIoU, below the default adaptation. Figure 3 illustrates selected variants on one sample; Figure 4 adds dataset-level evidence for the recovery order.

Dataset-level recovery order. We aggregate token reveal steps over each benchmark's evaluation examples and define boundary stabilization as the last update at which the boundary prediction changes. In Figure 4, s/S counts decoding progress forward, unlike the decreasing corruption index t in Section 3.

Without PT, semantic tokens emerge at 0.45–$0.47S$ on average, preceding time tokens at 0.69–$0.72S$. PT advances semantic reveal to 0.42–$0.44S$ and delays time-token reveal to 0.76–$0.79S$, increasing the within-dataset gap from 0.24–$0.25S$ to 0.34–$0.35S$. Boundary stabilization likewise shifts later, from 0.66–$0.68S$ to 0.72–$0.74S$. Together with the reversed-plan ablation, these results support semantic recovery before boundary determination beyond the single-sample trajectory. The means describe a dataset-level tendency, rather than an identical order for every example.

Loss Weights. Table 4 reports a local sweep on THUMOS-14 and ActivityNet-1.3 while keeping the 8B architecture and both-boundary gate fixed. The default setting $(1.0, 0.5, 0.2)$ performs best across the two benchmark families. Reducing λ_{plan} weakens the alignment between training corruption and boundary recovery. Increasing λ_{iou} overemphasizes local overlap optimization, and reducing λ_{ce} weakens the dense sequence supervision that couples labels with boundaries. The same setting also gives the strongest high-tIoU results, which indicates that the selected balance improves precise boundary placement rather than only coarse detection.

Figure 3: Illustrative single-sample denoising trajectories for selected component variants on ActivityNet-RTL. Temporal IoU is shown after a valid boundary pair first appears. Full MDVLM-TAL starts later and reaches higher final overlap than the weaker settings.

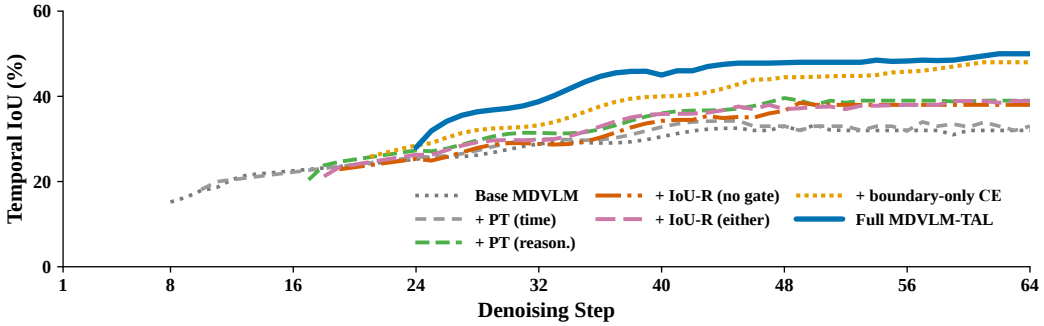
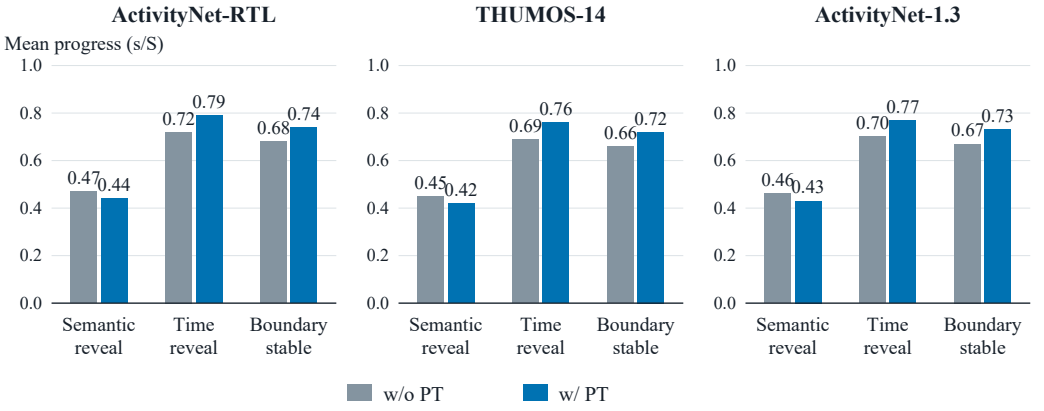


Figure 4: Dataset-level mean reveal and boundary-stabilization steps, with and without Planned Training (PT). Progress s/S counts forward through $S = 64$ decoding updates; larger values mean later recovery. Semantic tokens emerge earlier than time tokens on all three benchmarks, and PT increases this separation.



5 Conclusion

We introduced MDVLM-TAL, a masked diffusion vision-language model for temporal action localization that jointly denoises semantic reasoning tokens and temporal boundary tokens. Instead of predicting timestamps in a fixed left to right order, MDVLM-TAL keeps the full response editable throughout reverse denoising, so semantic evidence and temporal evidence can refine each other with bidirectional attention. This reframes temporal localization as a structured sequence-generation problem. In this view, localization quality depends not only on the final boundary tokens but also on how those tokens emerge during denoising.

From this perspective, we proposed some designs: a planned objective that keeps time tokens masked longer than semantic tokens and emphasizes lower-noise denoising steps, a step-level IoU reward that provides overlap-aware supervision during denoising, and a standard cross-entropy reconstruction term that stabilizes sequence generation. Our ablation results show that these components contribute in distinct but consistent ways: planned masking improves the denoising schedule, IoU-based reward improves boundary accuracy, and dense sequence supervision improves the coupling between semantics and localization. More broadly, MDVLM-TAL points toward a unified class of video-language models that treats localization, captioning, and reasoning as parts of the same vision-language generation process rather than as isolated modules.

Acknowledgments and Disclosure of Funding

This work was supported by the NSFC Regional Innovation and Development Joint Fund under Grant U25A20537, and the National Key Research and Development Program of China under Grant 2024YFC3015600.

References

[1] S. Bai, Y. Cai, R. Chen, K. Chen, X. Chen, Z. Cheng, L. Deng, W. Ding, C. Gao, C. Ge, et al. Qwen3-vl technical report. 2025. doi: 10.48550/arXiv.2511.21631.

[2] Y.-W. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, and R. Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. In *CVPR*, 2018.

[3] F. Cheng and G. Bertasius. Tallformer: Temporal action localization with a long-memory transformer. In *ECCV*, pages 503–521, 2022.

[4] Y. Feng, Z. Zhang, R. Quan, L. Wang, and J. Qin. Refinetad: learning proposal-free refinement for temporal action detection. In *ACM MM*, pages 135–143, 2023.

[5] Y. Guo, J. Liu, M. Li, D. Cheng, X. Tang, D. Sui, Q. Liu, X. Chen, and K. Zhao. Vtg-llm: Integrating timestamp knowledge into video llms for enhanced video temporal grounding. *AAAI*, 39(3):3302–3310, 2025. doi: 10.1609/aaai.v39i3.32341.

[6] F. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015.

[7] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[8] B. Huang, X. Wang, H. Chen, Z. Song, and W. Zhu. Vtimellm: Empower llm to grasp video moments. In *CVPR*, pages 14271–14280, 2024.

[9] D.-A. Huang, S. Liao, S. Radhakrishnan, H. Yin, P. Molchanov, Z. Yu, and J. Kautz. Lita: Language instructed temporal-localization assistant. In *ECCV*, pages 202–218. Springer Nature Switzerland, 2024.

[10] L. Huang, L. Wang, and H. Li. Weakly supervised temporal action localization via representative snippet knowledge propagation. In *CVPR*, pages 3272–3281, 2022.

[11] J. Kim, M. Lee, C.-H. Cho, J. Lee, and J.-P. Heo. Prediction-feedback detr for temporal action detection. *AAAI*, 39(4):4266–4274, 2025. doi: 10.1609/aaai.v39i4.32448.

[12] N. Kondapaneni, M. Marks, M. Knott, R. Guimaraes, and P. Perona. Text-image alignment for diffusion-based perception. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13883–13893. IEEE, 2024.

[13] P. Lee, J. Wang, Y. Lu, and H. Byun. Weakly-supervised temporal action localization by uncertainty modeling. *AAAI*, 35(3):1854–1862, 2021.

[14] K. Li, Y. He, Y. Wang, Y. Li, W. Wang, P. Luo, Y. Wang, L. Wang, and Y. Qiao. Videochat: chat-centric video understanding. *Science China Information Sciences*, 68(10), 2025. doi: 10.1007/s11432-024-4321-9.

[15] Q. Li, D. Liu, J. Kong, S. Li, H. Xu, and J. Wang. Temporal action localization with cross layer task decoupling and refinement. In *AAAI*, pages 4878–4886, 2025.

[16] S. Li, K. Kallidromitis, H. Bansal, A. Gokul, Y. Kato, K. Kozuka, J. Kuen, Z. Lin, K.-W. Chang, and A. Grover. Lavida: A large diffusion model for vision-language understanding. In *NeurIPS*, 2025.

[17] B. Liberatori, A. Conti, P. Rota, Y. Wang, and E. Ricci. Test-time zero-shot temporal action localization. In *CVPR*, pages 18720–18729, 2024.

[18] T. Lin, X. Liu, X. Li, E. Ding, and S. Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *ICCV*, pages 3889–3898, 2019.

[19] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[20] S. Liu, C. Zhao, F. Zohra, M. Soldan, A. Pardo, M. Xu, L. Alssum, M. Ramazanova, J. L. Alcázar, A. Cioppa, S. Giancola, C. Hinojosa, and B. Ghanem. Opentad: A unified framework and comprehensive study of temporal action detection. In *CVPR Workshops*, pages 2616–2626. IEEE, 2025.

[21] X. Liu, Q. Wang, Y. Hu, X. Tang, S. Zhang, S. Bai, and X. Bai. End-to-end temporal action detection with transformer. *IEEE Transactions on Image Processing*, 31:5427–5441, 2022.

[22] M. Maaz, H. Rasheed, S. Khan, and F. Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *ACL*, pages 12585–12602, 2024.

[23] S. Nag, X. Zhu, Y.-Z. Song, and T. Xiang. Zero-shot temporal action detection via vision-language prompting. In *ECCV*, pages 681–697. Springer, 2022.

[24] S. Nag, X. Zhu, Y.-Z. Song, and T. Xiang. Proposal-free temporal action detection via global segmentation mask learning. In *ECCV*, 2022.

[25] S. Nag, X. Zhu, J. Deng, Y.-Z. Song, and T. Xiang. Difftad: Temporal action detection with proposal denoising diffusion. In *CVPR*, 2023.

[26] S. Nie, F. Zhu, Z. You, X. Zhang, J. Ou, J. Hu, J. Zhou, Y. Lin, J.-R. Wen, and C. Li. Large language diffusion models. *Advances in Neural Information Processing Systems*, 38:50608–50646, 2026.

[27] D. Peng, Z. Zhang, P. Hu, Q. Ke, D. K. Yau, and J. Liu. Harnessing text-to-image diffusion models for category-agnostic pose estimation. In *European Conference on Computer Vision*, pages 342–360. Springer, 2024.

[28] T. Phan, K. Vo, D. Le, G. Doretto, D. Adjeroh, and N. Le. Zeetad: Adapting pretrained vision-language model for zero-shot end-to-end temporal action detection. In *WACV*, pages 7031–7040, 2024.

[29] Z. Qing, H. Su, W. Gan, D. Wang, W. Wu, X. Wang, Y. Qiao, J. Yan, C. Gao, and N. Sang. Temporal context aggregation network for temporal action proposal refinement. In *CVPR*, pages 485–494, 2021.

[30] H. Ren, W. Yang, T. Zhang, and Y. Zhang. Proposal-based multiple instance learning for weakly-supervised temporal action localization. In *CVPR*, pages 2394–2404, 2023.

[31] S. Reza, Y. Zhang, M. Moghaddam, and O. Camps. Hat: History-augmented anchor transformer for online temporal action localization. In *ECCV*, pages 205–222. Springer, 2024.

[32] D. Shi, Y. Zhong, Q. Cao, J. Zhang, L. Ma, J. Li, and D. Tao. React: Temporal action detection with relational queries. In *ECCV*, 2022.

[33] D. Shi, Y. Zhong, Q. Cao, L. Ma, J. Li, and D. Tao. Tridet: Temporal action detection with relative boundary modeling. In *CVPR*, pages 18857–18866, 2023.

[34] Y. Song, D. Kim, M. Cho, and S. Kwak. Online temporal action localization with memory-augmented transformer. In *ECCV*, pages 74–91. Springer, 2024.

[35] J. Tan, J. Tang, L. Wang, and G. Wu. Relaxed transformer decoders for direct action proposal generation. In *ICCV*, 2021.

[36] T. N. Tang, K. Kim, and K. Sohn. Temporalmaxer: Maximize temporal context with only max pooling for temporal action localization. 2023. doi: 10.48550/arXiv.2303.09055.

[37] F. Wang, J. Han, and Z. Tu. Recognition-conditioned reasoning: A training-free multimodal-llm pipeline for fine-grained micro-action understanding. In *Proceedings of the 34th ACM International Conference on Multimedia*, page 17293–17300, 2026.

[38] Y. Wang, K. Li, Y. Li, Y. He, B. Huang, Z. Zhao, H. Zhang, J. Xu, Y. Liu, Z. Wang, et al. Internvideo: General video foundation models via generative and discriminative learning. 2022. doi: 10.48550/arXiv.2212.03191.

[39] Y. Wang, K. Li, X. Li, J. Yu, Y. He, G. Chen, B. Pei, R. Zheng, Z. Wang, Y. Shi, et al. Internvideo2: Scaling foundation models for multimodal video understanding. In *ECCV*, pages 396–416. Springer, 2024.

[40] Y. Wang, X. Li, Z. Yan, Y. He, J. Yu, X. Zeng, C. Wang, C. Ma, H. Huang, J. Gao, et al. Internvideo2.5: Empowering video mllms with long and rich context modeling. 2025. doi: 10.48550/arXiv.2501.12386.

[41] J. Xiao, Y. Xiong, J. Han, and Z. Tu. Adaptive prototype model for attribute-based multi-label few-shot action recognition. *Pattern Recognition*, page 114784, 2026.

[42] L. Xu, Y. Zhao, D. Zhou, Z. Lin, S. K. Ng, and J. Feng. Pllava: Parameter-free llava extension from images to videos for video dense captioning. 2024. doi: 10.48550/arXiv.2404.16994.

[43] M. Xu, C. Zhao, D. S. Rojas, A. Thabet, and B. Ghanem. G-tad: Sub-graph localization for temporal action detection. In *CVPR*, 2020.

[44] M. Xu, M. Soldan, J. Gao, S. Liu, J.-M. Pérez-Rúa, and B. Ghanem. Boundary-denoising for video activity localization, 2023.

[45] M. Xu, M. Gao, Z. Gan, H.-Y. Chen, Z. Lai, H. Gang, K. Kang, and A. Dehghan. Slowfast-llava: A strong training-free baseline for video large language models. 2024. doi: 10.48550/arXiv.2407.15841.

[46] J. Yang, P. Wei, Z. Ren, and N. Zheng. Gated multi-scale transformer for temporal action localization. *IEEE Transactions on Multimedia*, 26:5705–5717, 2024.

[47] J. Yang, P. Wei, and N. Zheng. Cross time-frequency transformer for temporal action localization. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(6):4625–4638, 2024.

[48] S. Yu, J. Cho, P. Yadav, and M. Bansal. Sevila: Self-chained image-language model for video localization and question answering, 2023.

[49] Y. Zeng, Y. Zhong, C. Feng, and L. Ma. Unimd: Towards unifying moment retrieval and temporal action detection. In *ECCV*, pages 286–304. Springer Nature Switzerland, 2024.

[50] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer. Sigmoid loss for language image pre-training. In *ICCV*, pages 11975–11986, 2023.

[51] Y. Zhai, L. Wang, W. Tang, Q. Zhang, N. Zheng, D. Doermann, J. Yuan, and G. Hua. Adaptive two-stream consensus network for weakly-supervised temporal action localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4136–4151, 2023.

[52] C.-L. Zhang, J. Wu, and Y. Li. Actionformer: Localizing moments of actions with transformers. In *ECCV*, pages 492–510. Springer, 2022.

[53] H. Zhang, X. Li, and L. Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. In *EMNLP Demo*, pages 543–553. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-demo.49.

[54] Q. Zhang, J. Fang, R. Yuan, X. Tang, Y. Qi, K. Zhang, and C. Yuan. Weakly supervised temporal action localization via dual-prior collaborative learning guided by multimodal large language models. In *CVPR*, pages 24139–24148. IEEE, 2025.

[55] Z. Zhang, L. Xu, D. Peng, H. Rahmani, and J. Liu. Diff-tracker: text-to-image diffusion models are unsupervised trackers. In *European Conference on Computer Vision*, pages 319–337. Springer, 2024.

[56] Z. Zhang, Y. Zhou, D. Peng, J.-H. Lim, Z. Tu, D. W. Soh, and L. G. Foo. Visual prompting for one-shot controllable video editing without inversion. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7784–7794. IEEE, 2025.

[57] Z. Zhang, M. H. Huang, Z. Tu, and M.-H. Yang. Dart: Difficulty-adaptive routing for zero-shot video temporal grounding. In *European Conference on Computer Vision*, pages 73–91. Springer, 2026.

[58] Z. Zhang, Z. Tu, J. Yuan, D. W. Soh, and B. Du. Leveraging text-to-image diffusion models for unsupervised visual object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

[59] W. Zhao, Y. Rao, Z. Liu, B. Liu, J. Zhou, and J. Lu. Unleashing text-to-image diffusion models for visual perception. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5706–5716. IEEE, 2023.

[60] J. Zhou, L. Huang, L. Wang, S. Liu, and H. Li. Improving weakly supervised temporal action localization by bridging train-test gap in pseudo labels. In *CVPR*, pages 23003–23012, 2023.

[61] Y. Zhu, G. Zhang, J. Tan, G. Wu, and L. Wang. Dual detrs for multi-label temporal action detection. In *CVPR*, pages 18559–18569. IEEE, 2024. doi: 10.1109/CVPR52733.2024.01756.

[62] Z. Zhu, W. Tang, L. Wang, N. Zheng, and G. Hua. Enriching local and global contexts for temporal action localization. In *ICCV*, pages 13516–13525, 2021.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the masked diffusion TAL formulation, Planned Training Objective, Step-Level IoU Reward, and the evaluated benchmarks, which match the method and experiment sections.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper scopes its claims to ActivityNet-1.3, THUMOS-14, and ActivityNet-RTL, and the appendix discusses benchmark, compute, discretization, and backbone-dependence limitations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper is empirical and does not present formal theorems or proof-based theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The experiments section specifies datasets, prompts, metrics, backbones, timestamp discretization, optimizer settings, LoRA settings, sequence lengths, denoising steps, and loss weights.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper uses public benchmarks and cited models, and reports the experimental protocol needed to reproduce the main comparisons; code and data release details should be provided with the final artifact if applicable.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The main experimental settings are described in the implementation details, including model backbones, frame sampling, timestamp tokens, optimizer, training length, and inference configuration.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The paper reports single-run benchmark tables and ablations, but does not include error bars, confidence intervals, or repeated-seed statistical tests.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The experiments section states that all experiments are conducted on 4 NVIDIA L40S GPUs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [N/A]

   Justification: The work uses public video benchmarks and does not introduce human-subject experiments, privacy-sensitive data collection, or deployment on users.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

Justification: The paper studies temporal localization in videos and does not make deployment claims; possible societal impacts are limited to downstream video understanding applications.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release a deployed system or safety-critical model, so misuse safeguards are not applicable to this submission.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper credits the existing datasets, model backbones, and LoRA method through citations, including ActivityNet, LITA, LLaDA, SigLIP, and LoRA.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce or release a new dataset, benchmark, codebase, or model asset as part of the submission.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or direct interaction with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

Justification: The paper does not involve human-subject research, so IRB approval is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.