READ More than What You See: Reinforcement Learning for Accurate and Coherent Audio Description Generations

Anonymous ACL submission

Abstract

Audio Description aims to generate concise narrations of essential visual content in audiovisual media for blind and low-vision audiences. Existing methods either rely on prompting off-the-shelf multimodal models, which often mismatch AD style, or partially optimize training-based systems with next-token prediction, which under-explores model capacity and biases generation toward generic expressions. We present READ, the first reinforcement-learning (RL) framework for training-based AD generation. READ formulates AD as sequence-level optimization with reference-matching, length, and format rewards, and further introduces a dedicated coherence reward under context-aware supervision to promote narratively coherent descriptions. Experiments on MAD-Eval, CMD-AD, and TV-AD show that READ substantially outperforms prior methods across diverse evaluation metrics. Our results highlight RL as a promising paradigm for accurate and coherent AD generation. Our codes, models, and benchmark results will be publicly available.

1 Introduction

Audio Descriptions (ADs) refer to the generation of concise natural-language narrations that convey essential visual content in audiovisual media for blind and low-vision audiences. Unlike dense video captioning (Krishna et al., 2017), ADs are tightly constrained by narrative contexts and delivery time, as descriptions must be inserted around dialogues without disrupting the original audio (Pavel et al., 2020). High-quality AD therefore requires both **accurate** selection of salient plot-relevant visual information such as actions, expressions, and scene changes, and **coherent** narration that integrates naturally with contexts and story understanding.

Recent advances in large vision-language models (VLMs) (Wang et al., 2024; Bai et al., 2025; Hurst et al., 2024) have rapidly pushed forward automatic AD generation, giving rise to two main paradigms: *training-free* prompting and *training-based* models. The former rely on off-the-shelf multimodal models with carefully designed prompts and contextual cues (Zhang et al., 2024; Xie et al., 2024, 2025), while the latter adapt models to AD domain, typically by learning a projector from CLIP (Radford et al., 2021) visual features into the language model space (Han et al., 2023b, 2024; Fang et al., 2025; Wang et al., 2025).

However, current methods still face clear limitations. *Training-free* approaches often remain mismatched to the style of ADs, and their performance is inherently limited without task-specific learning. *Training-based* methods, on the other hand, can generate more in-domain descriptions, but most simply learn a lightweight projector on top of frozen modules, leaving the underlying model capacity for AD **in**sufficiently exploited. More importantly, prior end-to-end methods are typically optimized solely with next-token prediction loss, which encourages imitation of frequent patterns in the training data and can bias the model toward generic expressions rather than truly informative descriptions. While such objectives may improve local reference matching, they do not directly optimize sequence-level AD properties, particularly accurate content specification and consistency with the surrounding contexts.

Our motivation comes from recent progress in RL-based post-training, which has strongly improved reasoning in LLMs beyond next-token prediction (Shao et al., 2024; Guo et al., 2025), and is increasingly being extended to multimodal visual understanding (Huang et al., 2025; Deng et al., 2025; Feng et al., 2025). We view audio description in a similar light: *generating ADs is not merely a token-level matching problem, but an open-ended generation task that involves non-trivial reasoning over which visual content is salient, how it should be verbalized in AD style, and how each description*

should remain compatible with the contexts. This motivates us to go beyond conventional supervised fine-tuning and investigate RL for AD generation.

We propose **READ**, a **RE**inforcement-learning based framework for automatic **A**udio **D**escription generation. At its core, READ treats AD as a sequence level optimization problem, rather than relying solely on token-level supervision. Accordingly, the baseline framework is driven by a reference-matching reward that encourages generated descriptions to remain faithful to human-authored ADs at the sequence level. In practice, we instantiate this objective with a ROUGE-based reward computed between sampled rollouts and reference ADs, which provides a simple yet effective learning signal for this open-ended generation task. To account for the temporal constraints of AD, we further introduce a length reward that encourages clip-appropriate word counts, so that generated descriptions can be delivered within the available narration window. Additionally, we add a format reward to regularize the output structure, requiring intermediate reasoning and final descriptions to be enclosed in `<think>` and `<answer>` tags, respectively. Together, these designs establish a strong RL baseline for AD.

To further promote narrative coherence, we extend READ beyond clip-level supervision by incorporating additional training instances of the form `{context,clip}`→`AD`, so that the model is encouraged to generate descriptions that are compatible with the surrounding ADs. Building on this formulation, we introduce a dedicated coherence reward that explicitly favors contextually coherent generations. Concretely, this reward is based on a lightweight AD-adapted LLM, which evaluates the conditional plausibility of a generated description given its context. To prevent trivial gains from simply repeating contextual words, the coherence signal is computed in a way that discounts overlapping n-grams with the context. Our coherence reward is applied only when the generation already achieves reasonable content fidelity, allowing READ to improve narrative consistency without sacrificing descriptive accuracy. Extensive experiments show that sequence-level RL enables READ to substantially surpass prior methods on conventional captioning metrics, while also yielding significant improvements on AD-specific evaluation measures.

To summarize, our contributions are three-fold:

- To the best of our knowledge, READ is the first training-based AD framework to incorporate reinforcement learning, establishing a new optimization paradigm for automatic AD generation.
- Beyond improving fidelity to human-authored references, READ further introduces a dedicated coherence reward into RL training to promote narratively coherent, context-aware ADs.
- Extensive evaluations on MAD-Eval (Soldan et al., 2022), CMD-AD (Han et al., 2024), and TV-AD (Xie et al., 2024) show that READ substantially and consistently outperforms prior training-free and training-based methods, achieving large gains across a wide range of evaluation metrics.

2 Related Work

AD generation. Early AD systems mainly relied on specialized authoring tools (Branje and Fels, 2012) and skilled human contributors. Increasing efforts have been made to AD research with the rapid development of powerful LLMs (Radford et al., 2019; Touvron et al., 2023; Grattafiori et al., 2024) and VLMs (Radford et al., 2021; Wang et al., 2024; Hurst et al., 2024).

For automatic AD generation, training-free approaches (Chu et al., 2024; Ye et al., 2024; Zhang et al., 2024; Xie et al., 2024, 2025) typically customize their AD outputs by prompting foundation models together with handcrafted pipelines and manually designed visual cues (characters, contexts, shots, *etc*). Yet, these methods still lag in performance due to noticeable domain discrepancy between general-purpose LLM corpora and ADs. Alternatively, training-based methods (Han et al., 2023b,a, 2024; Lin et al., 2024; Fang et al., 2025; Wang et al., 2025) usually adapt VLMs to AD by fine-tuning a lightweight connector that projects visual features into the LLM space and optimizing with next-token prediction. Despite better performance, their training objective still tends to favor generic and less informative descriptions. In contrast, our READ introduces RL into automatic AD generation, treating AD as an open-ended generation task that requires more effective optimization beyond standard next-token supervision.

RL for (vision-)language models. RL has been widely used for sequence-level optimization in tasks such as machine translation (Ranzato et al., 2015), summarization (Paulus et al., 2017), and image captioning (Rennie et al., 2017). More recently, RL post-training has played an increasingly impor-

tant role in LLMs (Ouyang et al., 2022; Guo et al., 2025) and multimodal models (Huang et al., 2025; Deng et al., 2025), substantially enhancing alignment, reasoning, and visual understanding. However, such advances have not been systematically explored for automatic AD. READ fills this gap by introducing GRPO-based RL into training-based AD generation.

Context and coherence in AD. Context-aware generation has long been studied in AD, where neighboring descriptions, shot-level cues, or broader video contexts are used to improve narrative continuity and distinctiveness (Han et al., 2023b; Lin et al., 2024; Xie et al., 2025; Khandelwal et al., 2025). UniAD (Wang et al., 2025) and DistinctAD (Fang et al., 2025) fine-tune VLMs over multiple consecutive clips to incorporate boarder temporal context. However, most existing works rarely optimizing contextual coherence as a training objective explicitly, which results superficial coherence by repeating contextual content. In contrast, READ introduces an explicit coherence reward for AD generation and combines it with anti-copy masking to encourage context-compatible yet distinctive descriptions.

3 Method

3.1 Preliminary

Recent advances in reinforcement-learning-based post-training have demonstrated strong effectiveness in improving complex reasoning behaviors in large language and multimodal models. In this work, we focus on the RL component and build our method on Group Relative Policy Optimization (GRPO) (Shao et al., 2024), a variant of Proximal Policy Optimization (PPO) (Schulman et al., 2017) that avoids training an additional value model and is thus more computationally efficient. For each input q, GRPO samples a group of responses $\{o_1, o_2, \cdots, o_n\}$ from the current policy model π_θ. A reward function assigns scalar rewards $\{R_1, R_2, \cdots, R_n\}$ to these responses, and the relative advantage of each response is estimated by normalizing its reward within the group:

$$A_i = \frac{R_i - \text{mean}(\{R_1, R_2, \cdots, R_G\})}{\text{std}(\{R_1, R_2, \cdots, R_n\})}. \quad (1)$$

This group-relative formulation encourages the policy to prefer responses with higher relative quality within each sampled group. GRPO then optimizes the policy using a clipped objective together with KL regularization toward a reference model, which stabilizes the training updates while preserving useful prior behaviors:

$$\begin{aligned}\mathcal{J}_{\text{GRPO}} = &\mathbb{E}_{q,\{o_i\}}\Big[\tfrac{1}{G}\sum\nolimits_{i=1}^{G}\big(\min\big(\tfrac{\pi_\theta(o_i|q)}{\pi_{\theta_{\text{old}}}(o_i|q)}A_i,\\ &\text{clip}(\tfrac{\pi_\theta(o_i|q)}{\pi_{\theta_{\text{old}}}(o_i|q)}, 1-\epsilon, 1+\epsilon)A_i\big) - \\ &\beta\mathbb{D}_{\text{KL}}(\pi_\theta \parallel \pi_{\text{ref}})\big)\Big],\end{aligned} \quad (2)$$

where G is the rollout group size, ϵ is the clipping threshold, π_{ref} is the reference policy, and $\pi_{\theta_{\text{old}}}$ denotes the old policy.

3.2 GRPO for AD Generation: A Baseline

Building on the GRPO framework introduced above, we first construct a RL baseline for automatic AD generation, which we refer to as READ-base (Fig. 1). The baseline is designed to capture three basic requirements: 1) *content fidelity to human-authored references*, 2) *structural compliance with the desired output format*, and 3) *temporal suitability for narration*. Accordingly, READ adopts three fundamental rewards, namely *accuracy*, *format*, and *length*. We describe them below.

Accuracy reward. The core reward in READ is designed to measure how well a sampled description matches the human-authored reference. Since AD generation is an open-ended language generation task, exact matching is overly restrictive and cannot adequately reflect partial correctness in content selection and phrasing. Following (Feng et al., 2025), we adopt a sequence-level reference reward based on the average of ROUGE score (Lin, 2004). Given a sampled response o_i and its corresponding GroundTruth AD g, the accuracy reward is computed as:

$$R_i^{\text{acc}} = \tfrac{1}{3}\big(\rho_1(o_i, g) + \rho_2(o_i, g) + \rho_L(o_i, g)\big), \quad (3)$$

where ρ_1, ρ_2, and ρ_L denote ROUGE-1, ROUGE-2, and ROUGE-L, respectively. This reward provides an effective and efficient supervision signal for encouraging descriptions that remain faithful to the salient visual content of the target clip.

Format reward. We impose a simple formatting constraint on model responses. Specifically, the model is required to place its intermediate content within `<think>...</think>` tags and the final AD output within `<answer>...</answer>` tags. This structured format standardizes model outputs and

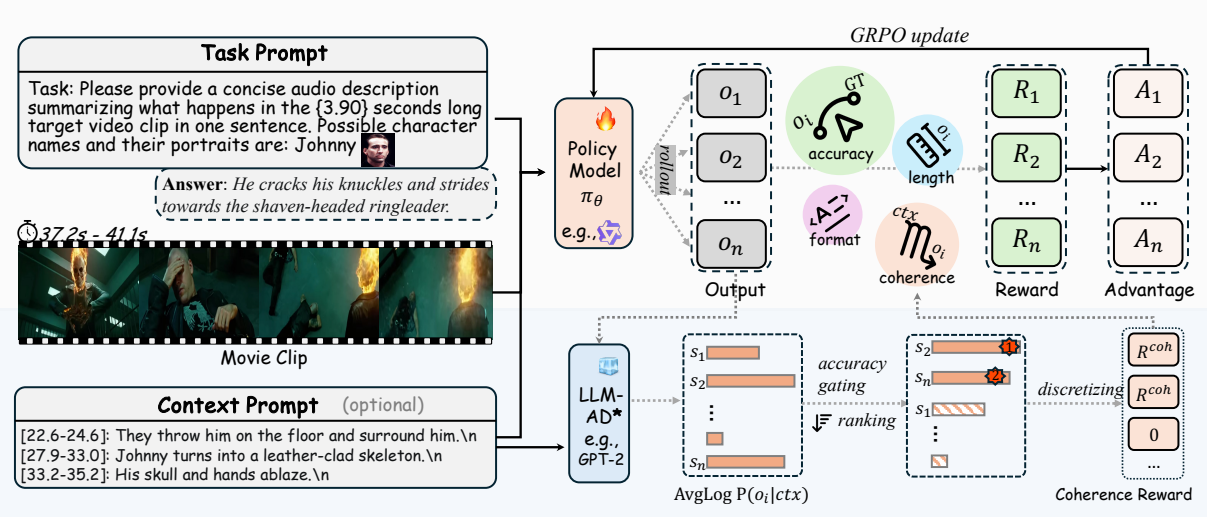

Figure 1: **Framework of READ.** For automatic AD generation, READ samples multiple candidates and optimizes them with GRPO using accuracy, format, and length rewards. It further introduces a context-aware coherence reward to encourage context-compatible and narratively coherent AD generation.

enables reliable extraction of the final AD description for reward computation. We therefore assign a positive reward R_i^{fmt} to a sampled response if it satisfies the required tagging structure.

Length Reward. Since movie clips vary substantially in duration, the appropriate length of the corresponding AD also differs across samples. Descriptions that are too short may omit important plot details, while overly long descriptions may exceed the available narration window and thus be difficult to deliver in time. We therefore introduce a length reward to encourage temporally appropriate AD generation. Concretely, we use the word count of the GroundTruth AD g as a simple proxy for the desired description length, and compare it with the word count of each sampled response. If the absolute length difference is within a predefined range, *i.e.*, $|\text{wc}(o_i) - \text{wc}(g)| \leq \Lambda$, the response is regarded as length-compatible and receives a positive reward R_i^{len}. (More analysis are in §C.1.)

3.3 Context-Aware Coherence Reinforcement

Although the READ-base improves clip-level fidelity, it optimizes each clip independently, leaving locally accurate outputs that may still be inconsistent with or redundant to surrounding ADs. To address this limitation, we introduce context-aware coherence reinforcement, which encourages each generated AD to remain faithful to the current clip while fitting the surrounding narrative context. Specifically, beyond original clip–AD pairs, we construct and merge context-aware training instances of the form $([c_1; \cdots; c_m], q) \rightarrow g$, where $[c_1, \ldots, c_m]$ are the temporal nearest contextual ADs associated with the current clip q. As shown in Fig. 1, the coherence reward is built as follows.

Since narrative coherence is meaningful only when a response is already reasonably faithful to the current clip, a sampled response is considered for coherence evaluation only if its ROUGE score (accuracy) exceeds a predefined threshold τ_{acc}. This ***gating*** strategy prevents the model from receiving extra reward for contextually plausible but visually inaccurate descriptions.

Coherence score. For each eligible response o_i, we concatenate its contextual ADs into a single context sequence $c_{1:m} = [c_1; \cdots; c_m]$, and feed $[c_{1:m}; o_i]$ into LLM-AD* to estimate the conditional likelihood of the response under the given context. LLM-AD* is a lightweight language model (*e.g.*, GPT-2) fine-tuned on AD corpora, making it better aligned with the concise narration style, temporal continuity commonly observed in ADs. (See §B.2 for more details.) We use it as a domain-adapted scorer to rank rollouts within the same group, rather than as an absolute evaluator of AD quality.

To avoid trivial gains from simply copying contextual content, we compute the coherence s_i using a masked average log-probability, where 3-gram or 4-gram spans of response tokens are excluded if they overlap with the context sequence. Formally, the coherence score of o_i is computed as

$$s_i = \frac{\sum_{t=1}^{L_i} m_{i,t} \log p_\phi(o_{i,t} \mid c_{1:m}, o_{i,<t})}{\sum_{t=1}^{L_i} m_{i,t}}, \quad (4)$$

where p_ϕ denotes the LLM-AD*, L_i is the response length, and $m_{i,t} \in \{0, 1\}$ is a token-level mask

indicating whether token $o_{i,t}$ is retained for scoring.

Coherence reward. We then rank all eligible responses within the same rollout group according to their coherence scores s_i in descending order. To make the reward signal more robust, we discretize the ranked scores into a small number of reward levels rather than using the raw language-model scores directly. Specifically, only the top-k ranked responses receive a positive coherence reward, with the reward magnitude determined by their rank level, while the remaining responses receive zero additional reward. The resulting coherence reward is defined as

$$R_i^{\text{coh}} = \begin{cases} \lambda_{\text{rank}(s_i)}, & \text{if } R_i^{\text{acc}} \geq \tau_{\text{acc}},\ \text{rank}(s_i) \leq k, \\ 0, & \text{otherwise}, \end{cases} \tag{5}$$

where $\lambda_{\text{rank}(s_i)}$ denotes a rank-dependent discretized coherence reward, k is the number of rewarded top-ranked responses.

3.4 Overall READ Training

Our final READ framework is trained by combining the four reward components described above into a unified reward:

$$R_i = R_i^{\text{acc}} + \lambda_1 R_i^{\text{fmt}} + \lambda_2 R_i^{\text{len}} + \lambda_3 R_i^{\text{coh}}, \tag{6}$$

where the accuracy reward R_i^{acc} serves as the anchor signal, and the remaining rewards are scaled accordingly. The resulting reward is used for policy optimization under the GRPO framework.

4 Experiment

4.1 Setup

Datasets. We evaluate READ on three standard datasets: CMD-AD (Han et al., 2024), MAD-Eval (Han et al., 2023b), and TV-AD (Xie et al., 2024). For CMD-AD, we train and evaluate on its official split, which contains ∼101K ADs from 1,432 movies, with 100 movies reserved for evaluation. For MAD-Eval, we follow the common protocol of training on MAD-v2-Named (Soldan et al., 2022), which contains ∼330K ADs from 488 movies, and evaluating on MAD-Eval, which consists of 6,520 ADs from 10 movies. For TV-AD, we use the official testing split with 3K evaluation ADs collected from 13 TV series.

Evaluation metric. We report conventional captioning metrics including CIDEr (Vedantam et al., 2015), ROUGE-L (Lin, 2004), METEOR (Banerjee and Lavie, 2005), BLEU-1 (Papineni et al., 2002), SPICE (Anderson et al., 2016) and AD-oriented evaluation metrics including Recall@k/N (Han et al., 2023a), Action (Xie et al., 2025) and LLM-AD-Eval (Han et al., 2024). Detailed introductions are in §B.3.

Implementation details. We instantiate READ on Qwen3-VL-8B (Bai et al., 2025) as the main backbone and fine-tune GPT-2 model (Radford et al., 2019) on pure AD corpus as LLM-AD*. For coherence RL, we construct and select context-aware training samples `{context,clip}`→`AD` according to their average temporal proximity. The ratio between these instances and standard `clip`→`AD` pairs is set to 3:1. Based on the true scale of accuracy reward R^{acc}, the weights for the $R^{\text{fmt}}, R^{\text{len}}, R^{\text{coh}}$ are set to 0.08, 0.05, and 0.05, respectively. The accuracy threshold for activating coherence is set to $\tau_{\text{acc}} = 0.12$. For rank-based coherence discretization, the top-2 responses in each rollout group receive additional coherence rewards. Detailed experimental settings can be seen in §B.

4.2 Comparisons with previous methods

We conduct comprehensive comparisons with prior training-free and training-based methods on widely-adopted MAD-Eval, CMD-AD and TV-AD.

Comparisons on MAD-Eval are shown in Tab. 1. Very early AD systems including ClipCap (Mokady et al., 2021), ClipDec (Nukrai et al., 2022), and MM-Vid (Lin et al., 2023) are omitted due to space limitation. We group previous studies into *Training-free* (upper rows) and *Training-based* (bottom rows) approaches, as described in Sec. 1. Existing training-based methods mainly adapt frozen visual encoders (CLIP) and LLMs (Touvron et al., 2023; Grattafiori et al., 2024) through lightweight projector tuning, whereas READ performs full-parameter RL training of the underlying VLMs (Wang et al., 2024; Bai et al., 2025). This allows the model to exploit its capacity more effectively for AD generation. With rule-based RL optimization, READ consistently outperforms all previous methods on every reported metric. Using Qwen3-VL-8B as backbone, READ obtains striking gains on conventional captioning metrics including BLEU-1 (22.7 *vs.* 15.9), CIDEr (40.0 *vs.* 28.2) and ROUGE-L (20.8 *vs.* 17.6). For AD-specific metrics, READ achieves the best R@5/16 (61.7 *vs.* 56.4) and Action score (36.1 *vs.* 33.5), indicating that the generated ADs are not only closer to human references but also more distinctive and semantically precise. These large and consistent improvements confirm

Method	VLM	LLM	Train?	BLEU-1	METEOR	ROUGE-L	SPICE	CIDEr	R@5/16	Action*
MM-Narrator	CLIP-L14	GPT-4	✗	12.8	6.7	13.4	5.2	13.9	49.0	–
LLM-AD	GPT-4V	–	✗	–	–	13.5	–	20.5	–	–
AutoAD-Zero	Qwen2-VL-7B	LLaMA3-8B	✗	13.6	6.6	14.6	7.8	23.6	51.3	–
AutoAD-Zero	GPT-4o	GPT-4o	✗	13.7	6.7	14.3	8.1	25.4	54.3	26.7
Shot-by-Shot	Qwen2-VL-7B	LLaMA3-8B	✗	16.2	7.2	14.7	7.8	25.0	50.6	27.9
Shot-by-Shot	GPT-4o	GPT-4o	✗	15.9	7.4	15.0	8.5	26.9	56.4	33.5
AutoAD-I	CLIP-B32	GPT-2	✓	–	–	11.9	4.4	13.4	42.1	–
AutoAD-II	CLIP-B32	GPT-2	✓	–	–	13.4	–	19.5	51.3	–
AutoAD-III	EVA-CLIP	LLaMA2-7B	✓	13.1	5.5	13.9	6.1	24.0	52.8	–
MovieSeq	CLIP-B16	LLaMA2-7B	✓	–	–	15.5	7.0	24.4	51.6	–
DistinctAD	CLIP_{AD}-B16	LLaMA3-8B	✓	–	–	17.6	8.3	27.3	56.0	–
Uni-AD	CLIP-L14	LLaMA3-8B	✓	–	–	17.2	–	28.2	54.9	–
READ (Ours)	Qwen2-VL-7B		✓	20.7	9.4	20.4	9.8	39.5	60.2	35.6
READ (Ours)	Qwen3-VL-8B		✓	**22.7**	**10.4**	**20.8**	**10.9**	**40.0**	**61.7**	**36.1**

Table 1: **Quantitative comparisons on MAD-Eval.** Best, second best, and third best denote the top-3 results for each metric. * denotes Action scores evaluated using the official Shot-by-Shot implementation (Xie et al., 2025).

Method	Train?	CMD-AD				TV-AD			
		CIDEr	Action	R@1/5	LLM-AD-Eval	CIDEr	Action	R@1/5	LLM-AD-Eval
Video-LLaMA	✗	4.8	–	22.0	1.89 | –	–	–	–	–
Video-BLIP	✗	5.2	–	23.6	1.91 | –	–	–	–	–
AutoAD-Zero	✗	22.4	30.7	32.9	3.08 | 2.49	30.9	26.8	34.7	3.12 | 2.57
Shot-by-Shot	✗	26.1	32.5	36.5	3.17 | **2.66**	34.2	27.4	36.6	3.12 | 2.59
AutoAD-II	✓	13.5	–	26.1	2.08 | –	–	–	–	–
AutoAD-III	✓	25.0	31.5	31.2	2.89 | 2.01	26.1	26.4	30.1	2.78 | 1.99
Uni-AD	✓	21.8	–	–	2.92 | –	26.6	–	–	2.89 | –
DistinctAD	✓	22.7	–	33.0	2.88 | 2.03	27.4	–	32.1	2.89 | 2.00
READ (Ours)	✓	**33.7**	**34.9**	**38.0**	**3.24** | 2.55	**40.9**	**30.7**	**40.0**	**3.16 | 2.62**

Table 2: **Quantitative comparison on CMD-AD and TV-AD.** Best, second best, and third best denote the top-3 results for each metric. For both evaluations, we report results with Qwen3-VL-8B as the backbone.

the effectiveness of READ on AD generation.

Among training-free approaches (Chu et al., 2024; Zhang et al., 2024; Xie et al., 2024, 2025), Shot-by-Shot achieves the strongest performance through careful shot threading, handcrafted text processing, and prompting, even surpassing most prior training-based methods. However, compared with prompt-heavy training-free pipelines, READ consistently outperforms these works by clear margins, highlighting the stronger potential of training-based AD systems for domain adaptation and practical deployment.

Comparisons on CMD-AD and TV-AD. We further examine the generalizabilty of READ on two extra benchmarks, with results reported in Tab. 2. Consistent with the results on MAD-Eval, READ also achieves the best overall performance on AD-oriented metrics. On CMD-AD, READ improves previous best CIDEr, Action, and R@1/5 scores from 26.1, 32.5, and 36.5 to 33.7, 34.9, and 38.0, respectively, and also obtains the highest LLM-AD-Eval score with LLaMA2 (3.24). READ likewise ranks first on all metrics on TV-AD, demonstrating strong generalizability from different perspectives. One exception is the second LLM-AD-Eval score evaluated by GPT-3.5 on CMD-AD, READ performs slightly lower than Shot-by-Shot (2.55 *vs.* 2.66). A possible reason is that Shot-by-Shot uses GPT-4o to generate ADs, which may enjoy an advantage when evaluated by a GPT-series judge. Nevertheless, READ remains clearly superior on the core reference-based and semantic metrics, showing that RL with our reward design yields substantial gains in both descriptive fidelity and action-level understanding.

Quantitative comparisons on coherence. Unlike CoherentAD (Khandelwal et al., 2025), which assesses coherence indirectly through story recall and repetition-based measures, we directly quantify coherence using our tuned LLM-AD scorer (Fig. 1, bottom) via the average conditional log-probability in Eq. 4. As shown in Tab. 3, the full READ model with R^{coh} achieves the best LogP (-3.61) and Masked LogP (-3.96), as well as the lowest PPL (99.5). Removing the coherence reward leads to consistent degradation on all metrics, confirming that our coherence reinforcement effectively improves contextual compatibility between

Method	Train?	LogP↑	Masked LogP↑	PPL↓
AutoAD-Zero	✗	-4.40	-4.57	273.6
Shot-by-Shot	✗	-5.21	-5.28	553.0
AutoAD-III	✓	-3.91	-4.32	121.0
READ-Base (w/o R^{coh})	✓	-4.19	-4.56	222.8
READ (w/ R^{coh})	✓	**-3.61**	**-3.96**	**99.5**

Table 3: **Quantitative coherence evaluation on CMD-AD.** LogP denotes the average log-probability of generated ADs conditioned on previous ADs, Masked LogP (Eq. 4) denotes the corresponding anti-copy masked average log-probability. PPL is short for perplexity.

#.	Setting	MAD-Eval			
		CIDEr	BLEU-1	R@5/16	Action
A.0	Qwen3-VL-8B (0-shot)	4.1	9.7	55.3	**38.0**
A.1	Qwen3-VL-8B (SFT)†	30.8	15.5	56.5	33.3
A.2	Baseline (R^{acc})	*Non-compliant outputs*			
A.3	Baseline ($R^{\text{acc}}, R^{\text{fmt}}$)	37.2	**24.1**	57.6	28.1
A.4	READ ($R^{\text{acc}}, R^{\text{fmt}}, R^{\text{len}}$)	38.5	21.6	61.0	35.4
A	**READ (w/ R^{coh})**	**40.0**	22.7	**61.7**	36.1
#.	Setting	CMD-AD			
		CIDEr	Action	R@1/5	LLM-Eval*
B.0	Qwen3-VL-8B (0-shot)	3.0	**36.8**	33.9	2.73
B.1	READ ($R^{\text{acc}}, R^{\text{fmt}}, R^{\text{len}}$)	32.4	32.1	36.5	3.01
B	**READ (w/ R^{coh})**	**33.7**	34.9	**38.0**	**3.24**

Table 4: **Ablation analysis.** † denotes supervised fine-tuning with next-token prediction. * denotes LLM-AD-Eval with LLaMA2-7B-Chat.

generated ADs and their preceding descriptions. This trend also suggests that training-free methods, *i.e.*, AutoAD-Zero and Shot-by-Shot, are generally weaker at maintaining narrative continuity, likely because they do not explicitly model context.

4.3 Ablation analysis

Training mechanism. We first verify the effectiveness of RL for AD generation (Tab. 4). On both benchmarks, RL-based READ markedly outperforms the zero-shot and SFT baselines, with the most notable gains on CIDEr, *e.g.*, from 30.8 to 40.0 over SFT (Tab. 4 row A.1 *vs.* A), and from 3.0 to 33.7 over zero-shot (B.0 *vs.* B). These results show that RL provides a much more effective training mechanism than direct prompting or standard next-token supervised fine-tuning. Although the zero-shot model achieves high Action scores, its outputs are overly detailed and lengthy, which increases the chance of matching GT verbs but tends to dilute salient information under other metrics; such behavior is also undesirable in practice, since ADs must fit within limited narration window between dialogues.

Baseline rewards. Next, we study the effect of major rewards in READ, on top of the anchor R^{acc}. Without format R^{fmt} (A.2), the model often fails to follow the required template, making it difficult to evaluate consistently. In A.3, removing length R^{len} encourages *reward hacking*, with the model favoring short and generic descriptions. While such outputs reduce the risk of obvious errors, they are also less informative, which is evidenced by the lowest Action (28.1) among all variants. Together, these three reward components establish a robust and effective READ baseline (A.4 & B.1).

Method	Coherence		CMD-AD			
	gating	*masking*	CIDEr	Action	R@1/5	LLM-Eval
Baseline	✗	✗	32.4	32.1	36.5	3.01
R^{coh} w/ gating	✓	✗	31.7	32.3	34.7	3.02
R^{coh} w/ masking	✗	✓	30.0	33.6	35.7	2.97
READ	✓	✓	**33.7**	**34.9**	**38.0**	**3.24**

Table 5: **Coherence reward design.**

Effectiveness of R^{coh}. We finally evaluate the effect of the coherence reward by comparing A.4 *vs.* A on MAD-Eval and B.1 *vs.* B on CMD-AD in Tab. 4. Despite R^{coh} is mainly designed to improve coherence with preceding ADs, it also consistently yields descent and stable gains on standard AD quality metrics. This indicates that, under the accuracy-gating design, coherence RL does not compromise descriptive fidelity, but instead helps the model refine its outputs into descriptions that are context-compatible and narratively coherent.

Coherence reward mechanism. *Accuracy gating* and *anti-copy masking* are important in coherence reward assignment (Tab. 5). The gating mechanism restricts coherence reinforcement to responses sufficiently faithful to the current clip, making coherence optimization more controlled. Anti-copy masking prevents the model from obtaining coherence reward by simply repeating contexts. Without masking, R@1/5 drops markedly (34.7), indicating reduced distinctiveness and clear reward hacking.

4.4 Training curves

Fig. 2 presents the reward curves during RL training. All curves show stable and reasonable trends, indicating that the optimization process is well-behaved. The format and length rewards increase rapidly and quickly plateau, suggesting that the model soon learns to follow the required output format and control the description length. The accuracy reward rises more gradually and saturates around 0.18, reflecting steady improvement in reference alignment. Although the coherence reward remains much smaller in magnitude due to the accuracy-gating mechanism and top-k reward assignment, it also shows a clear upward trend, indicating progressively better contextual coherence.

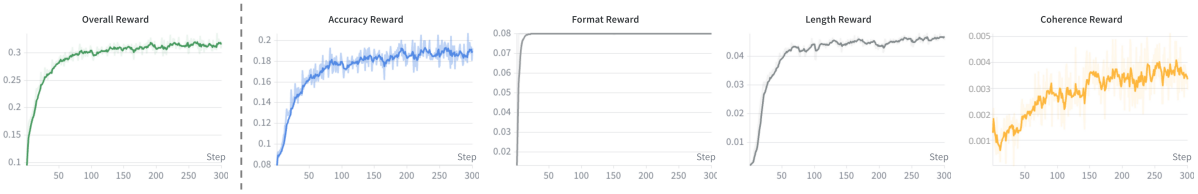

Figure 2: **Reward curves during RL training.**

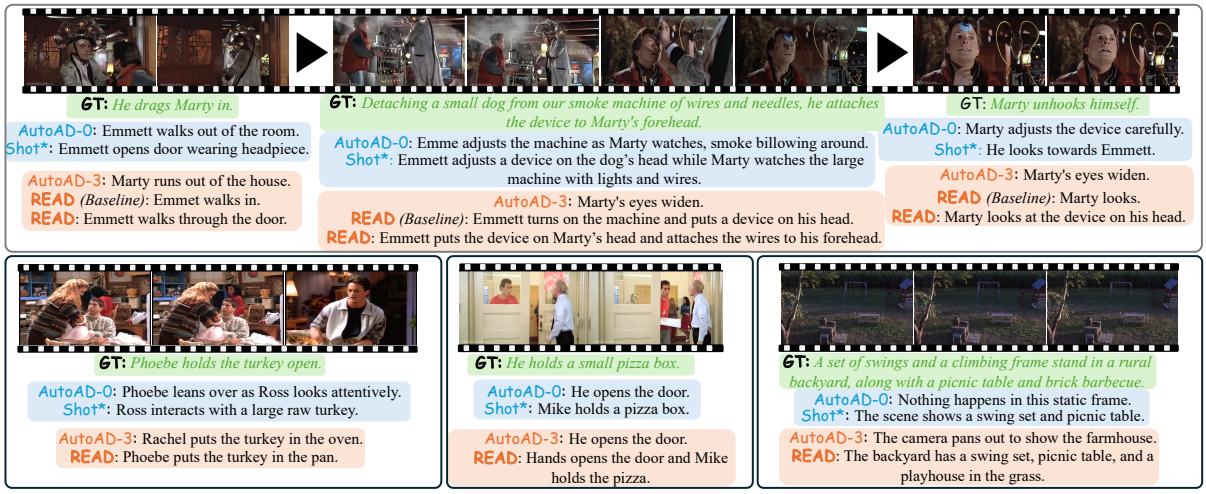

Figure 3: **Qualitative comparison of AD results.** GT is the ground-truth. Shot* is short for Shot-by-Shot. *Training-free* and *training-based* methods are highlighted seperately. Video frames are taken from *Back to the Future* (1985), *Friends* (S3E2), *Fast Times at Ridgemont High* (1982), and *Signs* (2002). Zoom in for details.

#.	Context	w/ R^{coh}?	CIDEr	Action	R@1/5	LLM-Eval*
A	NA	✗	32.4	32.1	**36.5**	3.01
A.1	Reccurrent	✗	32.2↓	31.3↓	35.5↓	2.99↓
A.2	Oracle	✗	**39.7**	**32.4**	34.9	**3.07**
B	NA	✓	33.7	34.9	**38.0**	3.24
B.1	Reccurrent	✓	34.2↑	35.2↑	36.1↓	3.25↑
B.2	Oracle	✓	**39.6**	**36.4**	34.2	**3.30**

Table 6: **Context evaluation on CMD-AD.** Arrows indicate whether performance improves or degrades relative to the recurrent setting. * denotes LLM-AD-Eval with LLaMA2-7B-Chat.

4.5 Evaluation under contextual settings

Since READ is trained with extra context-aware samples, it can naturally operate in a *recurrent* evaluation setting, where previous AD predictions are used as contexts (Han et al., 2023b). As shown in Tab. 6, coherence modeling is crucial in this setting: w/ R^{coh}, recurrent evaluation brings additional gains (B.1 *vs.* B), whereas w/o R^{coh}, introducing context instead degrades performance. With GT contexts (*oracle*), CIDEr improves substantially, whereas R@1/5 drops, indicating reduced distinctiveness. This suggests that the model may over-attend on the context and reproduce contextual content, rather than generating sufficiently discriminative ADs for the current clip.

4.6 Qualitative results

Fig. 3 shows qualitative comparisons of AD generation. In the top row, consisting of three *consecutive* clips, READ baseline (w/o R^{coh}) is more visually accurate than prior methods, while the full READ produces more coherent cross-clip descriptions. For instance, in the last clip, READ explicitly connects the current output to the previously mentioned "device," demonstrating better contextual continuity. In the bottom row, which contains three independent clips, READ generates ADs that are consistently closer to GT, further confirming its superiority. More examples are in §D.

5 Conclusion

We presented READ, an effective RL framework for automatic AD generation. READ introduces RL into training-based AD generation and further improves narrative coherence through a dedicated coherence reward with accuracy gating and anti-copy masking. Extensive experiments on multiple AD benchmarks show that READ consistently outperforms prior training-free and training-based methods across a wide range of metrics. These results demonstrate that reinforcement learning provides an effective training paradigm for generating accurate, distinctive, and context-compatible ADs.

Limitations

The limitations of READ includes: First, coherence modeling in READ relies on an external LLM-AD scorer and heuristic designs such as accuracy gating, reward discretization, and anti-copy masking, which may not fully capture higher-level narrative coherence. In addition, the current framework mainly uses local textual context from neighboring ADs, without explicitly modeling longer-range story structure, dialogue semantics, or audio cues. Finally, our experiments are conducted on existing English AD benchmarks, and the generalization of READ to broader domains, longer contexts, and other languages remains to be explored.

References

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In *European conference on computer vision*, pages 382–398. Springer.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, and 1 others. 2025. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Carmen J Branje and Deborah I Fels. 2012. Livedescribe: can amateur describers create high-quality audio description? *Journal of Visual Impairment & Blindness*, 106(3):154–165.

Peng Chu, Jiang Wang, and Andre Abrantes. 2024. Llm-ad: Large language model based audio description system. *arXiv preprint arXiv:2405.00983*.

Yihe Deng, Hritik Bansal, Fan Yin, Nanyun Peng, Wei Wang, and Kai-Wei Chang. 2025. Openvlthinker: An early exploration to complex vision-language reasoning via iterative self-improvement. *arXiv e-prints*, pages arXiv–2503.

Bo Fang, Wenhao Wu, Qiangqiang Wu, Yuxin Song, and Antoni B Chan. 2025. Distinctad: Distinctive audio description generation in contexts. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13571–13581.

Kaituo Feng, Manyuan Zhang, Hongyu Li, Kaixuan Fan, Shuang Chen, Yilei Jiang, Dian Zheng, Peiwen Sun, Yiyuan Zhang, Haoze Sun, and 1 others. 2025. Onethinker: All-in-one reasoning model for image and video. *arXiv preprint arXiv:2512.03043*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Tengda Han, Max Bain, Arsha Nagrani, Gul Varol, Weidi Xie, and Andrew Zisserman. 2023a. Autoad ii: The sequel-who, when, and what in movie audio description. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13645–13655.

Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman. 2023b. Autoad: Movie description in context. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18930–18940.

Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman. 2024. Autoad iii: The prequel-back to the pixels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18164–18174.

Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Xu Tang, Yao Hu, and Shaohui Lin. 2025. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Eshika Khandelwal, Junyu Xie, Tengda Han, Max Bain, Arsha Nagrani, Andrew Zisserman, Gül Varol, and Makarand Tapaswi. 2025. More than a moment: Towards coherent sequences of audio descriptions. *arXiv preprint arXiv:2510.25440*.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 706–715.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Kevin Lin, Faisal Ahmed, Linjie Li, Chung-Ching Lin, Ehsan Azarnasab, Zhengyuan Yang, Jianfeng Wang, Lin Liang, Zicheng Liu, Yumao Lu, and 1 others. 2023. Mm-vid: Advancing video understanding with gpt-4v (ision). *arXiv preprint arXiv:2310.19773*.

Kevin Qinghong Lin, Pengchuan Zhang, Difei Gao, Xide Xia, Joya Chen, Ziteng Gao, Jinheng Xie, Xuhong Xiao, and Mike Zheng Shou. 2024. Learning video context as interleaved multimodal sequences. In *European Conference on Computer Vision*, pages 375–396. Springer.

Ron Mokady, Amir Hertz, and Amit H Bermano. 2021. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*.

David Nukrai, Ron Mokady, and Amir Globerson. 2022. Text-only training for image captioning using noise-injected clip. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4055–4063.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*.

Amy Pavel, Gabriel Reyes, and Jeffrey P Bigham. 2020. Rescribe: Authoring and automatically editing audio descriptions. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, pages 747–759.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7008–7024.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pages 1279–1297.

Mattia Soldan, Alejandro Pardo, Juan León Alcázar, Fabian Caba, Chen Zhao, Silvio Giancola, and Bernard Ghanem. 2022. Mad: A scalable dataset for language grounding in videos from movie audio descriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5026–5035.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Hanlin Wang, Zhan Tong, Kecheng Zheng, Yujun Shen, and Limin Wang. 2025. Contextual ad narration with interleaved multimodal sequence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8372–8383.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and 1 others. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Junyu Xie, Tengda Han, Max Bain, Arsha Nagrani, Eshika Khandelwal, Gül Varol, Weidi Xie, and Andrew Zisserman. 2025. Shot-by-shot: Film-grammar-aware training-free audio description generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16503–16513.

Junyu Xie, Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman. 2024. Autoad-zero: A training-free framework for zero-shot audio description. In *Proceedings of the Asian Conference on Computer Vision*, pages 2265–2281.

Xiaojun Ye, Junhao Chen, Xiang Li, Haidong Xin, Chao Li, Sheng Zhou, and Jiajun Bu. 2024. Mmad: Multimodal movie audio description. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 11415–11428.

Chaoyi Zhang, Kevin Lin, Zhengyuan Yang, Jianfeng Wang, Linjie Li, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2024. Mm-narrator: Narrating long-form videos with multimodal in-context learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13647–13657.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Appendix

The appendix is organized as follows:

- **Appendix A** reports the LLM usage statement.
- **Appendix B** provides experimental details, including *(B.1)* the training prompts, *(B.2)* the training details and hyperparameter settings for READ and the LLM-AD scorer, and *(B.3)* the definitions of the evaluation metrics.
- **Appendix C** presents additional analyses, including *(C.1)* AD length statistics, *(C.2)* a comparison between ROUGE and CIDEr as the accuracy reward, and *(C.3)* the effect of longer RL training.
- **Appendix D** provides additional qualitative visualizations.

A LLM Usage

LLMs were primarily used to polish the writing of this paper, including improving fluency, grammar, and presentation. We also used LLMs in a limited scope to assist with minor implementation refinement, mainly for code optimization related to parallel evaluation and inference efficiency. ***GPT-5.4*** *was the primary LLM used. The core research ideas, method design, experimental setup, and final conclusions were developed and verified by the authors.*

B Experimental Details

B.1 Training prompts

Listing 1: Prompt of {clip -> AD}

```
<image><image><image><image><image><image><image
    ><image>\n
Audio Descriptions (ADs) are written narrations
    that describe the visual elements of a video
    , including characters, actions,
    interactions, and scene settings, especially
     those not conveyed through dialogue. The
    video clip is 2.70 seconds long. Consider
    the clip duration when completing the
    following task. \n
Please provide a concise audio description
    summarizing what happens in the video clip
    in one sentence, focusing on key characters,
     actions, interactions, or important visual
    details.\n
Possible character portraits and their names are:
     <image> Jeff Kohlver.
```

READ uses two types of training prompts, as shown in Lst. 1 and 2. The first is a standard

Hyperparameter	Value
Algorithm	GRPO
Base model	Qwen3-VL-8B-Instruct
Training batch size	32
Rollout batch size	128
Rollout per prompt (n)	8
Learning rate	2×10^{-6}
Max prompt length	16,384
Max response length	4,096
Precision	bf16
Optimizer	AdamW bf16
Epochs	1
KL coefficient	1×10^{-2}

Table 7: **Key hyperparameters for READ training with the VeRL framework.**

Hyperparameter	Value
Base model	GPT-2
Max sequence length	1,024
Batch size per device	8
Gradient accumulation	8
Learning rate	1×10^{-4}
Epochs	1
Warmup ratio	0.03
Weight decay	0

Table 8: **Hyperparameters for GPT-2 fine-tuning on the AudioVault corpus.**

`clip`→`AD` prompt, while the second is a context-aware prompt for `{context,clip}`→`AD` training.

Listing 2: Prompt of {context, clip -> AD}

```
<image><image><image><image><image><image><image
    ><image>\n
Audio Descriptions (ADs) are written narrations
    that describe the visual elements of a video
    , including characters, actions,
    interactions, and scene settings, especially
     those not conveyed through dialogue.\n\n

You are given a video clip with its prior and
    chronologically ordered contexts.\n
Context:\n
[AD] The sprinkler system goes on.\n
[AD] He drags the ringleader to his feet.\n
[AD] Ghost Rider puts on the skinheads leather
    jackets.\n\n

Task:\n
Please provide a concise audio description
    summarizing what happens in the target video
     clip in one sentence. The video clip is
    2.75 seconds long. Consider the clip
    duration when summarizing ADs.
```

B.2 Training details

Tab. 7 summarizes the main hyperparameter settings for fine-tuning READ on Qwen3-VL-8B-Instruct with the VeRL framework (Sheng et al., 2025). The Qwen2-VL-7B-Instruct variant uses the same training configuration. All experiments are conducted on 8 NVIDIA A800 GPUs with 80GB memory each.

Tab. 8 reports the hyperparameter settings for fine-tuning the LLM scorer on AudioVault[1]. We use GPT-2 as the base model and train it with casual language model modeling: given the preceding three adjacent ADs as context, the model predicts the next AD. The resulting scorer is then used for coherence evaluation and reward construction in READ.

[1]https://audiovault.net

B.3 Evaluation metrics details

CIDEr measures the similarity between a generated description and reference descriptions using TF-IDF (Salton and Buckley, 1988) weighted n-gram matching. It emphasizes informative content words and phrases.

ROUGE-L is based on the longest common subsequence (LCS) between the generated description and the reference. It reflects sequence-level overlap in content and word order.

METEOR evaluates word-level alignment between the generated description and the reference using a harmonic mean of precision and recall, with additional flexibility such as stemming and synonym matching. It is designed to better capture semantic similarity than exact lexical overlap.

BLEU-1 measures unigram precision between the generated description and the reference, together with a brevity penalty. It mainly reflects local lexical overlap.

SPICE compares the semantic content of the generated description and the reference by converting them into scene-graph tuples, such as objects, attributes, and relations. It focuses on semantic agreement rather than surface-level wording.

Recall@k/N is a retrieval-based metric that evaluates whether a generated description can be matched to the correct GT AD among N temporally adjacent references. For each generated AD,

similarity scores with the neighboring GT ADs are computed using BERTScore (Zhang et al., 2019), and Recall@k is then measured within this local temporal window.

Action measures whether a generated description correctly captures the actions in the reference AD. It combines two components: a semantic similarity score computed with text embeddings, and a verb-matching score based on whether the predicted and GT descriptions contain the same action verbs. The final score is obtained by combining these two signals.

LLM-AD-Eval uses a LLM as a judge to assess the quality of a generated AD with respect to the GT AD. Following prior work, the LLM is prompted to score each prediction-reference pair on a scale from 1 to 5, where higher scores indicate better matching quality.

LogP denotes the average log-probability of the generated AD conditioned on the contextual ADs. It measures how likely the generated description is under the context according to the LLM-AD scorer.

Masked LogP refers to the corresponding masked average log-probability, where tokens that overlap with context-copying spans are excluded from the computation. It is designed to better reflect genuine contextual compatibility rather than superficial repetition.

PPL (perplexity) is computed from the average token-level log-probability and reflects the overall uncertainty of the scorer on the generated AD. Lower PPL indicates that the generated description is more probable under the given context.

C More Analysis.

C.1 AD Length Statistics.

Fig. 4 summarizes the statistics of AD word count and the corresponding clip duration on the three benchmarks. Two observations are particularly important.

First, most ADs are concentrated in the lower-left region of the scatter plots, indicating that short descriptions dominate the data distribution, while longer ADs form a relatively sparse long tail. This imbalance suggests that pure next-token prediction can easily bias the model toward short and generic outputs, making RL-based optimization more suitable for controlling generation behavior.

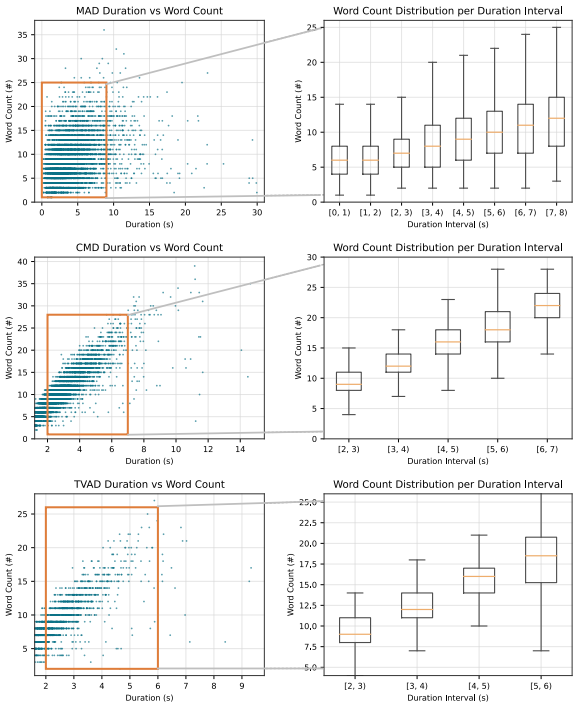

Figure 4: **Clip duration *vs.* AD word count statistics.**

Second, AD length is positively correlated with clip duration across all three datasets, showing that appropriate description length should depend on the temporal span of the target clip. This motivates the introduction of an explicit length reward in READ. Since the exact duration–length relationship varies across datasets, we do not fit a dataset-specific function; instead, we design the length reward based on the distance between the generated word count and the ground-truth AD length. In practice, the acceptable range is controlled by a tolerance threshold $\Lambda = 4$.

C.2 ROUGE vs. CIDEr as the Accuracy Reward

We further compare two choices for the anchor accuracy reward R^{acc}, namely the average ROUGE score in Eq. 3 and CIDEr. As shown in Tab. 9, using CIDEr as the optimization target yields slightly better performance on the final CIDEr metric itself, which is expected since the reward is more directly aligned with this evaluation measure. However, when coherence reinforcement is enabled, the ROUGE-based formulation produces consistently stronger AD-specific results, including higher Action, R@1/5, and LLM-AD-Eval scores.

A plausible explanation is that CIDEr and ROUGE provide different optimization signals. CIDEr is based on TF-IDF weighted n-gram match-

#.	Setting	R^{acc} Metric	BLEU-1	METEOR	ROUGE-L	SPICE	CIDEr	Action	R@1/5	LLM-AD-Eval
A.1	Baseline (w/o R^{coh})	**ROUGE** (Eq. 3)	18.4	7.2	16.5	7.5	32.4	**32.1**	36.5	3.01
A.2	Baseline (w/o R^{coh})	**CIDEr**	**19.9**	**7.6**	**17.0**	**7.6**	**35.1**	31.9	**37.4**	**3.05**
B.1	READ (w/ R^{coh})	**ROUGE** (Eq. 3)	19.7	**8.2**	17.5	7.7	33.7	**34.9**	**38.0**	**3.24**
B.2	READ (w/ R^{coh})	**CIDEr**	**20.9**	7.9	17.5	7.7	**35.8**	33.1	37.7	3.17

Table 9: **Effect of accuracy reward formulation on CMD-AD: ROUGE *vs.* CIDEr.**

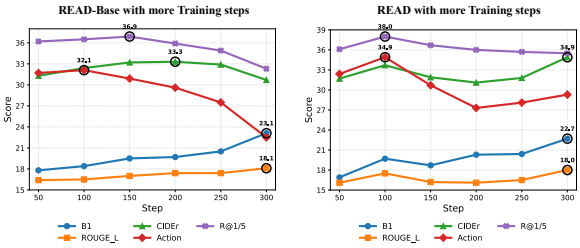

Figure 5: **Effect of longer RL training on READ-Base (left) and READ (right).**

ing and therefore more strongly favors exact overlap with informative reference phrases, which naturally benefits the final CIDEr metric. In contrast, the average ROUGE reward provides a smoother and more balanced sequence-level supervision signal, focusing more on overall content coverage and structural overlap. This makes it a more robust anchor reward when combined with coherence reinforcement: the model can rewrite its outputs to better fit the context without being overly penalized for moderate lexical variation. As a result, although CIDEr-based optimization yields slightly better CIDEr scores, the ROUGE-based formulation leads to stronger overall AD-specific performance.

C.3 Effect of Longer RL Training

Fig. 5 shows the effect of continuing RL optimization with more training steps for both READ-Base and READ. We observe that longer training further improves conventional captioning metrics such as BLEU-1, ROUGE-L, and CIDEr. However, this gain does not necessarily reflect better AD quality. Instead, it suggests a form of *reward hacking*: the model may increasingly rely on frequent words and generic expressions that are more likely to match the reference and thus obtain higher accuracy rewards. In contrast, AD-specific metrics such as Action and R@1/5 show a clear downward trend as training continues, indicating that the generated descriptions become less action-accurate and less distinctive. This discrepancy further highlights a limitation of using overlap-based metrics such as ROUGE or CIDEr as the anchor accuracy reward. How to design a more robust accuracy reward remains an important direction for future work.

D More Qualitative Results

We provide more qualitative results in Fig. 6. Compared with prior methods and the non-recurrent READ output, READ* better leverages narrative context to produce more coherent and context-compatible descriptions. The right side shows the intermediate thinking process, illustrating how READ reasons over (preceding AD context and) current visual content before generating the final description.

Figure 6: **Additional qualitative results.** READ* denotes recurrent inference using the previous three predicted ADs as context. The right side visualizes the intermediate thinking process used to infer the final description.