Think, then Score: Decoupled Reasoning and Scoring for Video Reward Modeling

Yuan Wang[1,2*§], **Ouxiang Li**[1 §], **Yulong Xu**[2‡], **Borui Liao**[2], **Jiajun Liang**[2†],
Jinghan Li[1], Meng Wang[2], Xintao Wang[2], Pengfei Wang[2], Kuien Liu[3], Xiang Wang[1†]
[1]University of Science and Technology of China, [2]Kling Team, Kuaishou Technology,
[3] Institute of Software Chinese Academy of Sciences
wy1001@mail.ustc.edu.cn, lioox@mail.ustc.edu.cn, liangjiajun@kuaishou.com

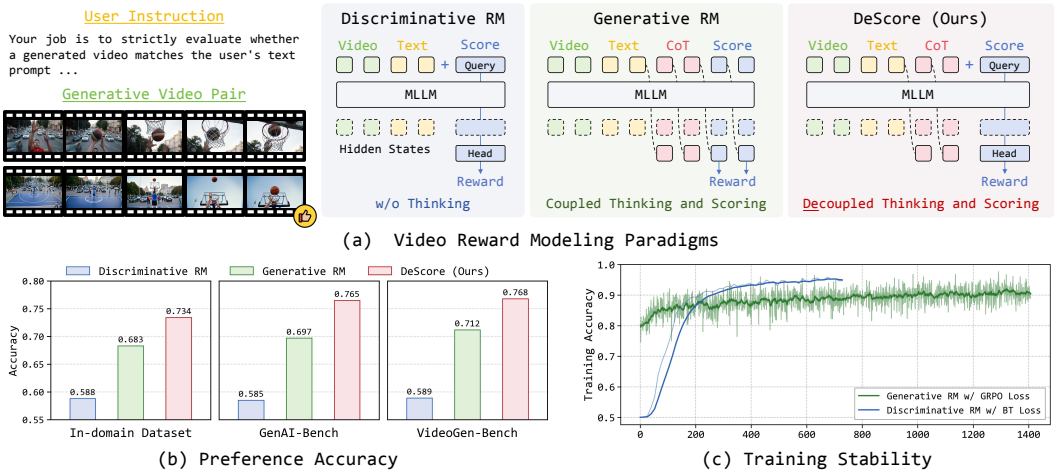


Figure 1: **Overview and Motivation of DeScore. (a) Video Reward Modeling Paradigms.** Existing video reward models generally follow two paradigms: Discriminative RMs directly regress rewards without explicit thinking (*e.g.*, CoT), and Generative RMs couple thinking and scoring within a single autoregressive sampling chain. DeScore improves both paradigms based on two observations: First, **(b) Preference Accuracy** shows that incorporating CoT enables Generative RMs to outperform Discriminative RMs, highlighting the necessity of explicit thinking for generalization. Second, **(c) Training Stability** reveals that coupling thinking and scoring requires the final score to be optimized through GRPO loss [7, 35], leading to pronounced training fluctuations. In contrast, discriminative training with BT loss [3] exhibits smooth convergence. Motivated by these findings, DeScore introduces a decoupled "think-then-score" paradigm that effectively leverages the generalization benefits of CoT reasoning while preserving the training stability inherent to discriminative optimization.

Abstract

Recent advances in generative video models are increasingly driven by post-training and test-time scaling, both of which critically depend on the quality of video reward models (RMs). An ideal reward model should predict accurate rewards that align with human preferences across diverse scenarios. However, existing paradigms face a fundamental dilemma: *Discriminative RMs* regress rewards directly on features extracted by multimodal large language models (MLLMs) without explicit reasoning, making them prone to shortcut learning and heavily reliant on massive data scaling for generalization. In contrast, *Generative RMs* with Chain-of-Thought

[§]Equal Contribution.
[†]Corresponding authors.
[‡]Project Lead.

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

(CoT) reasoning exhibit superior interpretability and generalization potential, as they leverage fine-grained semantic supervision to internalize the rationales behind human preferences. However, they suffer from inherent optimization bottlenecks due to the coupling of reasoning and scoring within a single autoregressive inference chain. To harness the generalization benefits of CoT reasoning while mitigating the training instability of coupled reasoning and scoring, we introduce DeScore, a training-efficient and generalizable video reward model. DeScore employs a decoupled "think-then-score" paradigm: an MLLM first generates an explicit CoT, followed by a dedicated discriminative scoring module consisting of a learnable query token and a regression head that predicts the final reward. DeScore is optimized via a two-stage framework: (1) a discriminative cold start incorporating a random mask mechanism to ensure robust scoring capabilities, and (2) a dual-objective reinforcement learning stage that independently refines CoT reasoning quality and calibrates the final reward, ensuring that higher-quality reasoning directly translates to superior model performance. Empirical evaluations demonstrate that DeScore achieves superior training efficiency and optimization stability, while outperforming state-of-the-art methods across diverse in-domain and out-of-distribution benchmarks. Moreover, DeScore also proves effective for post-training, leading to improved generated video quality. The code is available on: https://wyuan1001.github.io/DeScore/

1 Introduction

Modern generative video models [17, 23, 5, 25, 6, 39, 37, 16] have made remarkable progress in high-quality video synthesis, largely driven by post-training [20, 52, 38] and test-time scaling [22, 26]. Crucial to these paradigms is the video reward model, whose quality dictates the performance ceiling of the optimization process. An ideal video reward model must accurately align with human preferences across diverse scenarios and complex motion patterns. This necessitates robust out-of-distribution (OOD) generalization to maintain the accuracy of the reward signals.

One representative category of video reward models is the discriminative paradigm [10, 21], which typically regresses scalar rewards from multimodal large language model (MLLM) features. Despite the stable optimization signals offered by regression losses (*e.g.*, Bradley-Terry (BT) loss or MSE loss), the absence of explicit reasoning forces these models to infer fine-grained semantic differences from coarse preference labels. This often leads to shortcut learning [55], where models exploit shortcut features to fit training labels, rather than capturing the intrinsic semantic attributes aligned with human judgment. Compensating for this requires massive data scaling [21], which not only incurs prohibitive training overhead but also limits the model's adaptability to diverse OOD scenarios.

Another representative category follows the generative paradigm [47, 9, 45, 43, 44, 48], formulating reward modeling as a next-token prediction task within an MLLM framework. While directly generating a score token shares the limitations of discriminative models, advanced methods incorporate Chain-of-Thought (CoT) reasoning [47, 9, 45, 44] prior to the final reward. This process provides fine-grained semantic supervision, enabling the model to internalize the rationale behind human preferences. Specifically, the model learns why a video is superior rather than merely fitting a ranking, thereby enhancing its generalization potential, as evidenced by Figure 1 (b).

However, the generative paradigm predicts the reward as a token sequence in a next-token prediction manner rather than as an explicit scalar, which incurs the following optimization bottlenecks in Figure 1 (c). **(1) *Lack of direct reward-value optimization*:** Generative video reward models rely on supervised fine-tuning (SFT) and reinforcement learning (RL) for optimization. These methods fundamentally optimize discrete token probabilities instead of providing a direct gradient for the reward value, compared to the BT loss [3] (see Appendix A). Additionally, coupling logical reasoning and scoring within a single sampling chain forces a heavy reliance on RL (*e.g.*, GRPO [7, 35]) to improve performance, introducing two primary challenges: **(2) *Credit assignment difficulty*:** When an entire generated sequence shares a single rollout reward, it becomes difficult to determine whether a suboptimal output stems from low-quality intermediate reasoning tokens or an inaccurate final reward token. **(3) *High-variance policy gradients*:** RL-based policy optimization inherently suffers from high gradient variance [56, 11, 54], which leads to training instability (see Appendix B).

These challenges motivate a fundamental design question: ***How can we harness the interpretability and generalization introduced by CoT reasoning during reward modeling while shielding the training process from the optimization instability of a coupled sampling chain?*** To this end, we introduce DeScore, a training-efficient and generalizable video reward model through a decoupled "Think-then-Score" paradigm. By isolating reasoning from scoring, DeScore retains the fine-grained interpretability of generative CoT while mitigating the aforementioned bottlenecks through a specialized discriminative scoring module consisting of a learnable query token and a regression head. Crucially, this structural decoupling enables targeted optimization for the scoring module, bypassing the credit assignment dilemma caused by applying GRPO across the entire reasoning sequence. Moreover, the final scalar reward can be directly optimized via a stable, margin-based loss (*e.g.*, BT loss) rather than relying on high-variance policy gradients, thereby ensuring robust training efficiency.

To facilitate effective reward model training with our decoupled design, we instantiate DeScore based on Qwen3-VL-8B [2] and propose a two-stage training framework: (1) discriminative cold start and (2) dual-objective reinforcement learning (RL). In the cold-start stage, we jointly fine-tune the MLLM backbone and the scoring module using the BT loss. To improve robustness, we introduce a random masking mechanism that randomly drops the CoT during training. This strategy encourages the scoring module to leverage both the raw inputs and the generated CoT, preventing the reward prediction from being dominated by either source. During the RL stage, we employ a dual-objective optimization approach. The GRPO loss refines the reasoning quality of the CoT, while an auxiliary BT loss continuously calibrates the scoring module. This decoupled dual-objective explicitly isolates the reward optimization from the high-variance policy updates of the reasoning chain. By ensuring the scoring module receives a direct gradient for the reward value via the BT loss, DeScore achieves superior optimization stability and faster convergence while simultaneously refining the model's reasoning capabilities. Empirical evaluation shows that DeScore significantly outperforms existing discriminative and generative video reward models in terms of training efficiency and generalization performance. Our contribution can be summarized as follows:

- We introduce a decoupled video reward modeling paradigm that separates CoT reasoning from final reward prediction, combining the interpretability and generalization benefits of CoT reasoning, and maintains optimization stability and efficiency.
- We propose DeScore, a training-efficient and generalizable video reward model built on this paradigm with a two-stage framework: a discriminative cold start with random masking and a dual-objective RL stage that separates reasoning refinement from reward calibration.
- Extensive experiments demonstrate that DeScore consistently outperforms state-of-the-art (SOTA) baselines, achieving stronger OOD generalization and higher training efficiency. DeScore also proves effective for post-training, leading to improved generated video quality.

2 Related Work

Video Reward Model. Recent advances in visual generation [28, 29, 30, 50, 49, 51, 46], have significantly improved generation efficiency and controllability. Meanwhile, recent benchmarks [19] have revealed persistent challenges in compositional reasoning and instruction following, highlighting the growing need for reliable evaluation and reward modeling. Existing video reward models mainly follow two paradigms. Discriminative methods [10, 21] regress scalar rewards from MLLM features using objectives such as MSE or Bradley-Terry (BT) loss [3, 31]. Although these objectives provide stable optimization, the lack of explicit reasoning makes such models prone to shortcut learning [53] and heavily reliant on large-scale data to achieve strong generalization. Generative methods formulate reward modeling as next-token prediction. Early works [48, 45] directly generated scores, lacking the reasoning capacity to handle complex scenarios. Subsequent methods [43, 9, 44, 40] introduced CoT through two-stage training, *i.e.* SFT followed by RL [35]. Although CoT improves interpretability and generalization, these models often suffer from training instability because reasoning and scoring are coupled within a single sampling chain. Some methods [47] instead use token probabilities as rewards, but their reliance on reference videos or pairwise comparisons limits practical applicability. To address these limitations, we propose DeScore, which decouples reasoning from scoring through a "Think-then-Score" process, achieves both robust preference alignment and training efficiency.

Reinforcement Learning. Reinforcement learning (RL) has recently achieved strong performance across a wide range of MLLM tasks [13, 2, 14, 24, 8], substantially improving visual reasoning and understanding [57, 36, 41, 42]. Much of this progress has been driven by Group Relative Policy

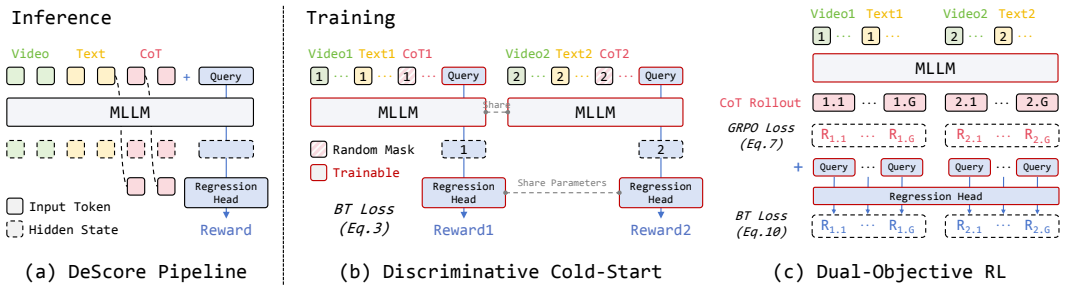


Figure 2: **Our DeScore framework.** (a) During inference, DeScore first uses an MLLM to generate CoT from the multi-modal input, then appends a learnable query token whose last hidden state is projected by a regression head into the final *video reward*. Training follows a two-stage paradigm: (b) In the discriminative cold-start stage, DeScore is trained with BT loss on pre-collected CoT data, where random CoT masking encourages the scoring module to use both multi-modal inputs and reasoning tokens. (c) In the dual-objective RL stage, the GRPO loss optimizes CoT rollouts guided by rule-based *rollout rewards*, while the BT loss simultaneously calibrates the *video reward*, decoupling reasoning refinement from reward scoring.

Optimization (GRPO) [35, 7], which estimates advantages from the relative rewards of multiple responses to the same input. By removing the need for a separate critic model [34], GRPO makes RL optimization more scalable for MLLMs. However, recent studies have identified important optimization bottlenecks in GRPO. [54] show that ineffective prompts can produce response groups that are uniformly correct or uniformly incorrect, thereby weakening effective gradient signals and increasing training variance. Meanwhile, [11] empirically show that the gradient variance of GRPO grows with sequence length, which leads training instability. These limitations are directly inherited by CoT-based video reward models [9, 43, 40], where reasoning and scoring are coupled in a single sampling chain, causing the final reward prediction to rely heavily on GRPO-based optimization. This motivates us to move beyond the standard GRPO objective and develop a more efficient optimization strategy for video reward modeling.

3 Method

3.1 Data Collection

We build our preference dataset by captioning diverse real-world videos and using the captions as prompts for multiple T2V models, including Gen-2 [32], Pika 1.0 [18], PixVerse (v1/v2) [27], Dreamina [4], Luma [1], Gen-3 [33], and Kling [17]. Human annotators compare the generated pairs along five alignment dimensions: object, dynamics, environment, style, and camera movement, resulting in 22K training pairs and 1469 in-domain evaluation pairs. We then generate stage-specific CoT annotations to support our two-stage training. Qwen3-VL-8B [2] is used for the discriminative cold-start stage to activate the scoring module, while Gemini-2.5-Pro [8] provides fine-grained CoTs with sub-dimension scores for the dual-objective RL stage. In both stages, we apply consistency-based filtering by retaining only CoTs whose implied preferences agree with human labels. Further details are provided in Appendix C.

3.2 Reward Model Learning

As illustrated in Figure 2, we propose DeScore, a decoupled reward modeling framework that achieves high generalization and training efficiency. DeScore uses Qwen3-VL-8B [2] as its multi-modal backbone, augmented with a scoring module comprising a learnable query token and a regression head. For a given generative video and the user instruction, the query token follows the generative CoT sequence, aggregating contextual information from multi-modal inputs and reasoning tokens via the MLLM backbone. Its hidden state is then projected by the regression head into a scalar reward. To ensure both reasoning quality and scoring accuracy, the optimization of DeScore adheres to a two-stage training paradigm. First, a discriminative cold start is performed on CoT data to enable the scoring module to effectively extract and aggregate semantic evidence from both multi-modal inputs and CoT, thereby yielding accurate scalar rewards. Subsequently, a dual-objective RL stage refines reasoning with the GRPO loss while calibrating the reward accuracy using the BT loss. User instructions for both training and inference are detailed in Appendix D.

Discriminative Cold Start. In this initial stage, our objective is to warm up the MLLM backbone and the scoring module, enabling them to effectively aggregate semantic information from both raw multi-modal inputs and pre-collected CoT data. Formally, given a generated video $\boldsymbol{v}$, a text prompt $\boldsymbol{c}$, and its corresponding pre-collected CoT $\boldsymbol{o}$, we construct the input sequence by appending a learnable query token `[Reward]` to the end:

$$\mathcal{X} = (\boldsymbol{v}, \boldsymbol{c}, \boldsymbol{o}, \texttt{[Reward]})\,. \tag{1}$$

The last hidden state of the `[Reward]` token, $\boldsymbol{h}_{\text{reward}} \in \mathbb{R}^d$, captures a condensed semantic summary of the multi-modal inputs and the reasoning process. It is then passed through the learnable regression head Φ to produce the scalar reward s:

$$s = \Phi(\boldsymbol{h}_{\text{reward}}). \tag{2}$$

To align the predicted rewards with human preferences, we employ the Bradley-Terry (BT) loss [3]. Given a preference pair from the dataset $\mathcal{D}$, consisting of a winning sample $\boldsymbol{q}^w = (\boldsymbol{v}^w, \boldsymbol{c}, \boldsymbol{o}^w)$ and a losing sample $\boldsymbol{q}^l = (\boldsymbol{v}^l, \boldsymbol{c}, \boldsymbol{o}^l)$, the model computes their respective scores s^w and s^l separately with our DeScore. The training objective is defined as:

$$\mathcal{L}_{\text{BT}} = -\mathbb{E}_{(\boldsymbol{q}^w, \boldsymbol{q}^l)\sim\mathcal{D}} \left[\log \sigma(s^w - s^l)\right], \tag{3}$$

where $\sigma(\cdot)$ denotes the sigmoid function.

To ensure that the decoupled scoring module effectively utilizes both the multi-modal video inputs and the generated CoT, preventing the module from relying solely on the CoT, we apply a random masking strategy during training. During training, the CoT $\boldsymbol{o}$ is masked with a probability p. In these instances, the reward s is computed solely based on the raw multi-modal inputs ($\boldsymbol{v}, \boldsymbol{c}, \boldsymbol{o} = \varnothing$). This mechanism forces DeScore to maintain a strong grounding in the original video features, ensuring that the final reward is a holistic reflection of both visual evidence and logical reasoning, thereby enhancing the robustness of the reward prediction.

Reinforcement Learning with Dual-Objective. In the second stage, we fine-tune the entire model using a dual-objective RL strategy. We employ Group Relative Policy Optimization (GRPO) to refine the reasoning quality of the CoT. However, optimizing solely for CoT quality can lead to "reward drift", where the scoring module loses its calibration. To mitigate this, we combine the GRPO objective with an auxiliary BT loss. During this stage, the model first generates a CoT reasoning sequence $\boldsymbol{o}$ conditioned on the input $\hat{\boldsymbol{q}} = (\boldsymbol{v}, \boldsymbol{c})$. The final input sequence for reward prediction is constructed as

$$\mathcal{X} = (\boldsymbol{v}, \boldsymbol{c}, \boldsymbol{o}, \texttt{[Reward]})\,. \tag{4}$$

The scalar reward s is then computed by passing $\mathcal{X}$ through the MLLM backbone Θ and the regression head Φ:

$$s = \Phi(\Theta(\mathcal{X})_{\texttt{[Reward]}}) = \Phi(\boldsymbol{h}_{\text{reward}}), \tag{5}$$

where $\boldsymbol{h}_{\text{reward}}$ is the hidden state of the `[Reward]` token, aggregating information from both the multi-modal inputs and the generated CoT $\boldsymbol{o}$.

Following the standard GRPO framework, we sample a group of G responses $\{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_G\}$ from the old policy $\pi_{\theta_{\text{old}}}$ for each input $\hat{\boldsymbol{q}}$. The advantage A_i for the i-th response is computed by normalizing the rewards within the group:

$$A_i = \frac{R(\boldsymbol{o}_i) - \text{mean}(\{R(\boldsymbol{o}_1), R(\boldsymbol{o}_2), \ldots, R(\boldsymbol{o}_G)\})}{\text{std}(\{R(\boldsymbol{o}_1), R(\boldsymbol{o}_2), \ldots, R(\boldsymbol{o}_G)\})}. \tag{6}$$

Let $\mathcal{Q}$ denote the human preference training set, the GRPO loss can be formulated as:

$$\begin{aligned}\mathcal{L}_{\text{GRPO}}(\theta) = -\,\mathbb{E}_{\hat{\boldsymbol{q}}\sim\mathcal{Q},\{\boldsymbol{o}_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\boldsymbol{o}|\hat{\boldsymbol{q}})} \Bigg\{ \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|\boldsymbol{o}_i|}\sum_{t=1}^{|\boldsymbol{o}_i|}\min\Bigg[\frac{\pi_\theta(o_{i,t}\mid\hat{\boldsymbol{q}},\boldsymbol{o}_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}\mid\hat{\boldsymbol{q}},\boldsymbol{o}_{i,<t})}A_i, \\ \text{clip}\left(\frac{\pi_\theta(o_{i,t}\mid\hat{\boldsymbol{q}},\boldsymbol{o}_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}\mid\hat{\boldsymbol{q}},\boldsymbol{o}_{i,<t})}, 1-\epsilon, 1+\epsilon\right)A_i\Bigg]\Bigg\} + \beta D_{\text{KL}}\left(\pi_\theta\|\pi_{\text{ref}}\right),\end{aligned} \tag{7}$$

where t indexes the token position in each generated response, $o_{i,t}$ denotes the t-th token of the i-th response $\boldsymbol{o}_i$, and $\boldsymbol{o}_{i,<t}$ denotes the preceding token sequence used as the autoregressive context. The

clipping threshold ϵ bounds the importance sampling ratio, while β controls the KL regularization strength to prevent the optimized policy π_θ from deviating excessively from the reference policy π_{ref}. To improve CoT generation, the composite reward $R(\boldsymbol{o}_i)$ is designed with three components:

$$R(\boldsymbol{o}_i) = \lambda_1 R_{\text{fmt}}(\boldsymbol{o}_i) + \lambda_2 R_{\text{qual}}(\boldsymbol{o}_i) + \lambda_3 R_{\text{len}}(\ell_i), \tag{8}$$

where λ_1, λ_2, and λ_3 are the trade-off weights, and $\ell_i = |\boldsymbol{o}_i|$ denotes the length of the generated CoT.

- *Format Reward* (R_{fmt}): Assigns 1 if the output strictly follows the `<think></think>` structure and provides a JSON-formatted sub-dimension score, otherwise 0.
- *Quality Reward* (R_{qual}): Measures the accuracy of the predicted sub-dimension scores against ground-truth labels: $R_{\text{qual}} = N_{\text{correct}}/N_{\text{total}}$.
- *Length Reward* (R_{len}): Encourages detailed reasoning while penalizing excessive verbosity or extreme brevity:

$$R_{\text{len}}(\ell) = \begin{cases} 0, & \ell < 500, \\ 0.2 \times \lfloor \ell/500 \rfloor, & 500 \le \ell < 2000, \\ 1, & \ell \ge 2000. \end{cases} \tag{9}$$

In addition to the GRPO objective, we apply an auxiliary BT loss to calibrate the final reward, ensuring that improvements in CoT quality consistently translate into gains in overall model performance. Given a training pair $(\hat{\boldsymbol{q}}^w, \hat{\boldsymbol{q}}^l)$ consisting of a winning and a losing sample, we generate their respective CoT rollouts, denoted as $\mathcal{O}^w = \{\boldsymbol{o}_1^w, \ldots, \boldsymbol{o}_G^w\}$ and $\mathcal{O}^l = \{\boldsymbol{o}_1^l, \ldots, \boldsymbol{o}_G^l\}$. While the GRPO loss is computed based on these rollouts, each response $\boldsymbol{o}_i^j$ (where $j \in \{w, l\}$) is also used to construct the input sequence $\mathcal{X}_i^j = \left(\hat{\boldsymbol{q}}^j, \boldsymbol{o}_i^j, \texttt{[Reward]}\right)$. We then compute the scalar reward s_i^j for each rollout according to Eq. 5. The final auxiliary BT loss is defined as:

$$\mathcal{L}_{\text{BT}}^{\text{aux}} = -\mathbb{E}_{(\hat{\boldsymbol{q}}^w, \hat{\boldsymbol{q}}^l)\sim\mathcal{D}} \left[\frac{1}{G} \sum_{i=1}^{G} \log \sigma(s_i^w - s_i^l) \right]. \tag{10}$$

To integrate dual objectives, the final training loss for this stage is formulated as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{GRPO}} + \alpha \mathcal{L}_{\text{BT}}^{\text{aux}}, \tag{11}$$

where α is a balancing coefficient to align gradient scales. This decoupled design ensures the final reward remains grounded in a stable regression rather than dominated by a coupled sampling chain.

3.3 Inference

During inference, DeScore evaluates videos via a two-step "think-then-score" procedure. Given a test generative video $\boldsymbol{v}$ and user prompt $\boldsymbol{c}$, the backbone Θ first autoregressively generates a detailed CoT $\boldsymbol{o}$ to analyze video quality. Subsequently, the query token `[Reward]` is appended to form the sequence $\mathcal{X}_{\text{inf}} = (\boldsymbol{v}, \boldsymbol{c}, \boldsymbol{o}, \texttt{[Reward]})$. The MLLM backbone Θ processes the input sequence $\mathcal{X}_{\text{inf}}$, and the resulting hidden state of the `[Reward]` token, $\boldsymbol{h}_{\text{reward}}$, is fed into the regression head Φ to produce the scalar reward s, integrating information from both the multi-modal inputs and the generated CoT:

$$s = \Phi(\Theta(\mathcal{X}_{\text{inf}})_{\texttt{[Reward]}}). \tag{12}$$

By decoupling scoring from reasoning, DeScore harnesses the interpretability and generalization introduced by CoT reasoning while shielding the training process from the optimization instability of a coupled sampling chain.

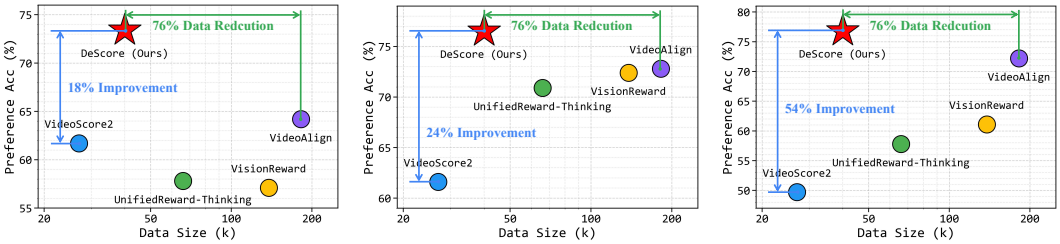
4 Experiments

4.1 Experimental Setups.

Implementation. We use Qwen3-VL-8B [2] as the backbone of DeScore. In the discriminative cold-start stage, the model is fine-tuned with LoRA (rank 64) for two epochs using AdamW, with a learning rate of 2×10^{-6}, weight decay of 0.01, and batch size of 32. The resulting checkpoint is then used to initialize the RL stage. In the RL stage, we optimize DeScore with GRPO and auxiliary BT losses, using coefficients of 1.0 and 0.005, respectively. GRPO is trained with a learning rate of

Table 1: **Main Experiments on Video Reward Benchmarks.** We evaluate preference accuracy on in-domain and OOD benchmarks. **Bold** and underlined values denote the best and second-best results. DeScore achieves the strongest overall performance, demonstrating superior generalization.

Model	In-domain Dataset	OOD Video Reward Benchmark			
		GenAI		VideoGen-Bench	
		Acc w/ Tie	Acc w/o Tie	Acc w/ Tie	Acc w/o Tie
Discriminative Video Reward Model					
VideoScore [10]	0.552	0.490	0.720	0.372	0.503
VideoAlign [21]	0.642	0.494	0.728	0.538	0.722
Generative Video Reward Model					
VisionReward [48]	0.571	0.525	0.724	0.465	0.611
UnifiedReward [45]	0.492	0.458	0.686	0.303	0.564
UnifiedReward-Thinking [43]	0.578	**0.548**	0.709	0.428	0.582
VideoScore2 [9]	0.617	0.391	0.616	0.301	0.497
Our Video Reward Model					
DeScore (Ours)	**0.734**	0.504	**0.765**	**0.568**	**0.768**



(a) In-domain Dataset (b) GenAI-Bench (c) VideoGen-Bench

Figure 3: **Performance vs. Training Data Size.** DeScore (red star) consistently outperforms existing models by a large margin while requiring only a fraction of the training data, highlighting its extreme training efficiency and robust semantic understanding.

1×10^{-6}, group size $G = 8$, 65 training steps, a rollout batch size of 128, and a mini-batch size of 32. Video inputs are processed at 2 fps during both training and inference, with more details shown in Appendix E.

Baselines and Benchmarks. We evaluate DeScore against discriminative baselines, including VideoScore [9] and VideoAlign [21], and generative baselines, including VisionReward [48], UnifiedReward [45], UnifiedReward-Thinking [43], and VideoScore2 [9], where the latter two generate CoT before the final reward. Experiments are conducted on an in-domain preference dataset with 1,469 pairs and two OOD benchmarks: GenAI [15], containing 1.9k low-resolution short-video pairs from early T2V models, and VideoGen-Bench [21, 55], containing 26.5k higher-resolution and longer-video pairs from current SOTA models. Detailed settings are provided in Appendix F.

4.2 Main Results

High Generalization. To evaluate the reward accuracy of DeScore, we compare it with several state-of-the-art baselines across both in-domain and OOD benchmarks. As shown in Table 1, DeScore outperforms both discriminative and generative methods on the in-domain preference dataset, and this advantage consistently extends to OOD settings. On GenAI-Bench, although UnifiedReward-Thinking achieves a slightly higher Acc w/ Tie, DeScore obtains the best result on the more informative Acc w/o Tie metric (0.765), demonstrating strong performance on videos generated by early-stage T2V models. On VideoGen-Bench, DeScore reaches 0.768 in Acc w/o Tie, substantially outperforming the strongest discriminative baseline, VideoAlign (0.722), and the best generative baseline (0.582). These results show that DeScore achieves state-of-the-art performance across most benchmarks, validating the effectiveness of our "think-then-score" paradigm in combining the interpretability of generative reasoning with strong generalization.

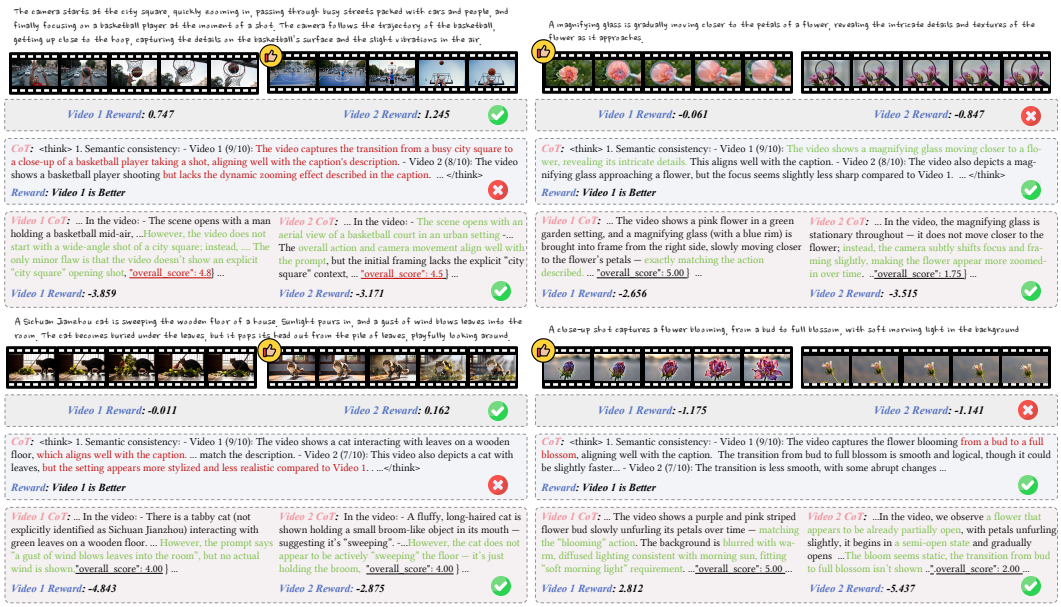


Figure 4: **Qualitative Comparison of Different Video Reward Models.** We compare VideoAlign, UnifiedReward-Thinking, and our DeScore with high- and low-quality reasoning within these responses. DeScore consistently yields accurate rewards and robust reasoning across varied prompts, demonstrating its superior interpretability and generalization.

Training Efficiency. As shown in Figure 3, DeScore achieves a superior performance-to-data trade-off across all evaluated benchmarks, outperforming SOTA reward models while reducing training data by 76%. These results show that the decoupled "think-then-score" paradigm improves sample efficiency by leveraging fine-grained semantic rationales, bypassing the heavy data dependency of traditional discriminative video reward models.

Qualitative Performance. As shown in Figure 4, we compare DeScore with two representative baselines, the discriminative model VideoAlign and the generative model UnifiedReward-Thinking, across different scenarios including camera motion (row 1) and dynamic motion (row 2). DeScore consistently performs well in all cases. In particular, UnifiedReward-Thinking often fails when the generated CoT is of low quality, since its final score is directly coupled with the reasoning path. In contrast, DeScore decouples reasoning from scoring and applies random masking during Stage 1, encouraging the scoring module to jointly leverage multimodal inputs and CoT rather than relying solely on generated text. This yields two clear advantages: (1) *error tolerance*, where DeScore remains accurate even with imperfect CoT (*e.g.* top-left case), and (2) *fine-grained discrimination*, where it can still produce differentiated scores when generative models output identical reward tokens (*e.g.* bottom-left case). These results show that DeScore effectively combines reasoning interpretability with robust reward prediction. We further compare the inference cost of DeScore with that of a SOTA generative reward model in Appendix **??**. The two models exhibit comparable decoding speeds, indicating that DeScore's additional inference overhead stems primarily from its independent reward-readout stage rather than slower autoregressive generation.

4.3 Ablation Study

To evaluate the contribution of each component and setting in DeScore, we perform ablation studies on our in-domain dataset and two OOD benchmarks. The results are summarized in Table 2.

Effectiveness of Stage 1 Components. Table 2 (top) illustrates the impact of the CoT and random masking mechanism during the discriminative cold-start phase. Incorporating CoT leads to a significant performance boost across all benchmarks, achieving an accuracy improvement of 2.7% (from 0.588 to 0.615) on the in-domain dataset, alongside substantial gains on OOD benchmarks (where Acc w/o Tie increases by 5.1% on GenAI and 6.5% on VideoGen-Bench). This confirms that explicit rationales improve semantic understanding and reward prediction. Random masking further improves performance on the in-domain dataset (0.615 $\rightarrow$ 0.656), GenAI (0.636 $\rightarrow$ 0.685), and

Table 2: **Ablation Study of DeScore Components across Different Training Stages.** We investigate the impact of CoT and Random Masking during Stage 1 (top), and the contributions of Cold Start and the auxiliary BT Loss to Stage 2 (bottom). **Bold** values denote the best performance in each stage.

Training Satge	Setting		In-domain Dataset	OOD Video Reward Benchmark			
				GenAI		VideoGen-Bench	
	CoT	Random Mask		Acc w/ Tie	Acc w/o Tie	Acc w/ Tie	Acc w/o Tie
Stage 1 (Discriminative Version)	×	×	0.588	0.379	0.585	0.427	0.589
Stage 1	✓	×	0.615	0.417	0.636	0.480	0.654
Stage 1 (Default)	✓	✓	**0.656**	**0.449**	**0.685**	**0.489**	**0.672**
	Setting		In-domain Dataset	GenAI		VideoGen-Bench	
	Cold Start	BT Loss		Acc w/ Tie	Acc w/o Tie	Acc w/ Tie	Acc w/o Tie
Stage 2 (Generative Version)	×	×	0.683	0.461	0.697	0.526	0.712
Stage 2	✓	×	0.691	0.471	0.754	0.445	0.648
Stage 2	×	✓	0.720	0.491	0.748	0.547	0.741
Stage 2 (Default)	✓	✓	**0.734**	**0.504**	**0.765**	**0.568**	**0.768**

VideoGen-Bench ($0.654 \rightarrow 0.672$). To validate our hypothesis that random masking encourages the model to jointly leverage multi-modal inputs and CoT, rather than over-relying on reasoning tokens, we visualize the top 150 tokens receiving the highest attention from the final reward query. As shown in Figure 6 in Appendix H.1, random masking causes the query token to attend to both multi-modal inputs and CoT, rather than relying solely on reasoning tokens with better reward performance.

Effectiveness of Stage 2 Components. Table 2 (bottom) evaluates the contributions of cold-start initialization and the dual-objective optimization in the reinforcement learning (RL) stage.We observe that training with only the GRPO loss leads to a significant performance drop, with accuracy on VideoGen-Bench declining from 0.768 to 0.648. We attribute this to a misalignment between reasoning and scoring. While GRPO improves reasoning quality, it neglects the scoring module, leading the model to sacrifice reward accuracy for better rationales. In contrast, introducing the auxiliary BT loss effectively calibrates the reward output. This ensures that improvements in CoT quality consistently translate into gains in overall model performance. Furthermore, it is worth noting that even without cold-start initialization, the dual-objective RL training still achieves respectable performance across benchmarks, though it remains slightly lower than the default setting. These results highlight the robustness of our design, confirming that the dual-objective training remains effective even without a highly optimized starting point.

Comparison on Reward Modeling Paradiagm. We compare DeScore with two representative variants: a discriminative version and a generative version, whose training details are provided in Appendix G. The discriminative version (Table 2, row 1) removes CoT and predicts rewards solely from multi-modal inputs using a regression head, similar to VideoAlign. The generative version (Table 2, bottom row 1) follows the standard two-stage pipeline of SFT and GRPO, and predicts rewards via next-token generation, similar to VideoScore2. Both variants underperform DeScore across all benchmarks. In particular, the discriminative variant performs notably worse, indicating limited generalization and a stronger reliance on data scaling to maintain accuracy. This is further supported by the efficiency analysis in Figure 3, which shows that it requires substantially more training data to achieve comparable performance. Figure 1 (c) further compares the optimization stability of the two variants, showing that training with BT loss yields a smoother and more consistent improvement in preference accuracy, whereas GRPO exhibits pronounced fluctuations.

4.4 Improving Video Generation

To further demonstrate the effectiveness of DeScore for improving video generation, we integrate it into two representative post-training frameworks, Longcat-GRPO [37] and Flow-DPO [21], built on Wan-2.1-1.3B [39], and evaluate the resulting models on VBench [12]. Detailed settings are provided in Appendix G. As shown in Table 3, DeScore consistently improves generation quality under both frameworks, yielding gains in subject consistency (SC), background consistency (BC), aesthetic quality (AQ), image quality (IQ), and dynamic degree

Table 3: **Comparison of Reward Models for Improving Video Generation Quality on VBench.** DeScore consistently improves the quality of generated videos under two different post-training paradigms.

Model	SC↑	BC↑	AQ↑	IQ ↑	DD ↑
Wan-2.1-1.3B	0.951	0.961	0.547	0.669	0.527
w/ Longcat-GRPO	0.969	0.973	0.645	0.706	0.541
w/ Flow-DPO	0.969	0.972	0.615	0.700	0.542

(DD). These results show that DeScore serves as an effective reward model for post-training higher-quality video generators. We further provide qualitative comparisons in Figure 7 of Appendix H.2, which illustrate that DeScore improves prompt fidelity and video quality across diverse scenarios.

5 Conclusion

In this work, we introduced DeScore, a decoupled "Think-then-Score" framework for video reward modeling. By decoupling CoT reasoning from final reward prediction, DeScore retains the interpretability and generalization benefits of explicit reasoning while avoiding the optimization instability caused by coupled reasoning and scoring in generative reward models. With a two-stage training framework, DeScore achieves stable and efficient optimization and consistently outperforms existing discriminative and generative baselines on both in-domain and OOD benchmarks, while requiring substantially less training data. Moreover, DeScore further improves generated video quality when applied to post-training and test-time scaling. These results highlight decoupled reasoning and scoring as a promising paradigm for training-efficient and generalizable video reward modeling.

6 Acknowledgements

This research is supported by National Natural Science Foundation of China (62625608, 62572449) and the Open Innovation Program for jointly trained master's-to-doctoral students at the Institute of Computing Technology, Chinese Academy of Sciences (2024-02-KF-LH- 13).

References

[1] Luma AI. Luma. `https://lumalabs.ai/dream-machine`, 2025.

[2] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[3] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[4] ByteDance. Dreamina. `https://dreamina.capcut.com/`, 2024.

[5] ByteDance. Seedream. `https://seed.bytedance.com/zh/seedream4_0`, 2025.

[6] ByteDance. Seedance. `https://seed.bytedance.com/zh/seedance2_0`, 2026.

[7] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025.

[8] Google. Gemini-2.5-pro. `https://deepmind.google/models/gemini/pro/`, 2025.

[9] Xuan He, Dongfu Jiang, Ping Nie, Minghao Liu, Zhengxuan Jiang, Mingyi Su, Wentao Ma, Junru Lin, Chun Ye, Yi Lu, et al. Videoscore2: Think before you score in generative video evaluation. *arXiv preprint arXiv:2509.22799*, 2025.

[10] Xuan He, Dongfu Jiang, Ge Zhang, Max Ku, Achint Soni, Sherman Siu, Haonan Chen, Abhranil Chandra, Ziyan Jiang, Aaran Arulraj, et al. Videoscore: Building automatic metrics to simulate fine-grained human feedback for video generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2105–2123, 2024.

[11] Zhiyuan He, Xufang Luo, Yike Zhang, Yuqing Yang, and Lili Qiu. Vl norm: Rethink loss aggregation in rlvr. *arXiv preprint arXiv:2509.07558*, 2025.

[12] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024.

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

[14] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

[15] Dongfu Jiang, Max Ku, Tianle Li, Yuansheng Ni, Shizhuo Sun, Rongqi Fan, and Wenhu Chen. Genai arena: An open evaluation platform for generative models. *Advances in Neural Information Processing Systems*, 37:79889–79908, 2024.

[16] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

[17] Kuaishou. Kling. `https://app.klingai.com/cn/`, 2026.

[18] Pika Labs. Pika. `https://pika.art`, 2023.

[19] Ouxiang Li, Yuan Wang, Xinting Hu, Huijuan Huang, Rui Chen, Jiarong Ou, Xin Tao, Pengfei Wan, Xiaojuan Qi, and Fuli Feng. Easier painting than thinking: Can text-to-image models set the stage, but not direct the play? In *International Conference on Learning Representations*, volume 2026, pages 86729–86758, 2026.

[20] Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *arXiv preprint arXiv:2505.05470*, 2025.

[21] Jie Liu, Gongye Liu, Jiajun Liang, Ziyang Yuan, Xiaokun Liu, Mingwu Zheng, Xiele Wu, Qiulin Wang, Menghan Xia, Xintao Wang, et al. Improving video generation with human feedback. *arXiv preprint arXiv:2501.13918*, 2025.

[22] Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, et al. Inference-time scaling for diffusion models beyond scaling denoising steps. *arXiv preprint arXiv:2501.09732*, 2025.

[23] MiniMax. Hailuo. `https://hailuoai.com/`, 2024.

[24] OpenAI. Gpt-5. `https://openai.com/index/introducing-gpt-5/`, 2025.

[25] OpenAI. Sora. `https://openai.com/zh-Hans-CN/sora/`, 2025.

[26] Yuta Oshima, Masahiro Suzuki, Yutaka Matsuo, and Hiroki Furuta. Inference-time text-to-video alignment with diffusion latent beam search. *arXiv preprint arXiv:2501.19252*, 2025.

[27] PixVerse. Pixverse. `https://pixverse.ai`, 2025.

[28] Junxiang Qiu, Lin Liu, Shuo Wang, Jinda Lu, Kezhou Chen, and Yanbin Hao. Accelerating diffusion transformer via gradient-optimized cache. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17608–17617. IEEE, 2025.

[29] Junxiang Qiu, Jinda Lu, and Shuo Wang. Multimodal generation with consistency transferring. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 504–513, 2025.

[30] Junxiang Qiu, Shuo Wang, Jinda Lu, Lin Liu, Houcheng Jiang, Xingyu Zhu, and Yanbin Hao. Accelerating diffusion transformer via error-optimized cache. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 9588–9597, 2025.

[31] Pejaver V Rao and Lawrence L Kupper. Ties in paired-comparison experiments: A generalization of the bradley-terry model. *Journal of the American Statistical Association*, 62(317):194–204, 1967.

[32] Runway. Gen-2. `https://runwayml.com/research/gen-2`, 2023.

[33] Runway. Gen-3. https://runwayml.com/research/introducing-gen-3-alpha, 2024.

[34] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

[35] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024.

[36] Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, Ruochen Xu, and Tiancheng Zhao. VLM-R1: A stable and generalizable r1-style large vision-language model. *CoRR*, abs/2504.07615, 2025.

[37] Meituan LongCat Team, Xunliang Cai, Qilong Huang, Zhuoliang Kang, Hongyu Li, Shijun Liang, Liya Ma, Siyu Ren, Xiaoming Wei, Rixu Xie, et al. Longcat-video technical report. *arXiv preprint arXiv:2510.22200*, 2025.

[38] Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8228–8238, 2024.

[39] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

[40] Qunzhong Wang, Jie Liu, Jiajun Liang, Yilei Jiang, Yuanxing Zhang, Yaozhi Zheng, Xintao Wang, Pengfei Wan, Xiangyu Yue, and Jiaheng Liu. Vr-thinker: Boosting video reward models through thinking-with-image reasoning. *arXiv preprint arXiv:2510.10518*, 2025.

[41] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *CoRR*, abs/2508.18265, 2025.

[42] Xiyao Wang, Chunyuan Li, Jianwei Yang, Kai Zhang, Bo Liu, Tianyi Xiong, and Furong Huang. Llava-critic-r1: Your critic model is secretly a strong policy model. *CoRR*, abs/2509.00676, 2025.

[43] Yibin Wang, Zhimin Li, Yuhang Zang, Chunyu Wang, Qinglin Lu, Cheng Jin, and Jiaqi Wang. Unified multimodal chain-of-thought reward model through reinforcement fine-tuning. *arXiv preprint arXiv:2505.03318*, 2025.

[44] Yibin Wang, Zhiyu Tan, Junyan Wang, Xiaomeng Yang, Cheng Jin, and Hao Li. Lift: Leveraging human feedback for text-to-video model alignment. *arXiv preprint arXiv:2412.04814*, 2024.

[45] Yibin Wang, Yuhang Zang, Hao Li, Cheng Jin, and Jiaqi Wang. Unified reward model for multimodal understanding and generation. *arXiv preprint arXiv:2503.05236*, 2025.

[46] Yuan Wang, Ouxiang Li, Tingting Mu, Yanbin Hao, Kuien Liu, Xiang Wang, and Xiangnan He. Precise, fast, and low-cost concept erasure in value space: Orthogonal complement matters. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28759–28768. IEEE, 2025.

[47] Jie Wu, Yu Gao, Zilyu Ye, Ming Li, Liang Li, Hanzhong Guo, Jie Liu, Zeyue Xue, Xiaoxia Hou, Wei Liu, et al. Rewarddance: Reward scaling in visual generation. *arXiv preprint arXiv:2509.08826*, 2025.

[48] Jiazheng Xu, Yu Huang, Jiale Cheng, Yuanming Yang, Jiajun Xu, Yuan Wang, Wenbo Duan, Shen Yang, Qunlin Jin, Shurun Li, et al. Visionreward: Fine-grained multi-dimensional human preference learning for image and video generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 11269–11277, 2026.

[49] Yiyan Xu, Qiulin Wang, Wenjie Wang, Yunyao Mao, Xintao Wang, Pengfei Wan, Kun Gai, and Fuli Feng. Unicustom: Unified visual conditioning for multi-reference image generation. *arXiv preprint arXiv:2605.12088*, 2026.

[50] Yiyan Xu, Wenjie Wang, Yang Zhang, Biao Tang, Peng Yan, Fuli Feng, and Xiangnan He. Personalized image generation with large multimodal models. In *Proceedings of the ACM on Web Conference 2025*, pages 264–274, 2025.

[51] Yiyan Xu, Ruoxuan Xia, Wuqiang Zheng, Fengbin Zhu, Wenjie Wang, and Fuli Feng. Nextads: Towards next-generation personalized video advertising. *arXiv preprint arXiv:2603.02137*, 2026.

[52] Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818*, 2025.

[53] Wenqian Ye, Guangtao Zheng, and Aidong Zhang. Rectifying shortcut behaviors in preference-based reward learning. *arXiv preprint arXiv:2510.19050*, 2025.

[54] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

[55] Ailing Zeng, Yuhang Yang, Weidong Chen, and Wei Liu. The dawn of video generation: Preliminary explorations with sora-like models. *arXiv preprint arXiv:2410.05227*, 2024.

[56] Junzi Zhang, Jongho Kim, Brendan O'Donoghue, and Stephen Boyd. Sample efficient reinforcement learning with reinforce. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 10887–10895, 2021.

[57] Yifan Zhang, Xingyu Lu, Xiao Hu, Chaoyou Fu, Bin Wen, Tianke Zhang, Changyi Liu, Kaiyu Jiang, Kaibing Chen, Kaiyu Tang, Haojie Ding, Jiankang Chen, Fan Yang, Zhang Zhang, Tingting Gao, and Liang Wang. R1-reward: Training multimodal reward model through stable reinforcement learning. *CoRR*, abs/2505.02835, 2025.

A Analysis on Optimization Direction

In generative reward models, the final reward, whether a point-wise score or a pair-wise preference, is formulated as a discrete token $s \in \mathcal{V}$. During the supervised fine-tuning (SFT) stage, optimization heavily relies on the cross-entropy (CE) loss for next token prediction, formulated as $\mathcal{L}_{CE} = -\log \pi_\theta(s^*|\hat{\boldsymbol{q}}, \boldsymbol{o})$, where s^* is the target score token. However, this formulation treats rewards as orthogonal classes, penalizing categorical mismatches while entirely ignoring ordinal distance during reward optimization (*e.g.*, mispredicting a 5 as a 4 incurs a similar penalty as predicting a 1). This fundamental limitation persists into the reinforcement learning (RL) stage. Optimization typically relies on policy gradients (*e.g.*, GRPO), where the gradient with respect to parameters θ is approximated as $\hat{g}_{GRPO} \propto A \cdot \nabla\theta \log \pi_\theta(s|\hat{\boldsymbol{q}}, \boldsymbol{o})$. Here, A represents the advantage, $\hat{\boldsymbol{q}}$ denotes the multi-modal inputs, and $\boldsymbol{o}$ represents the generated CoT. Critically, this categorical treatment lacks numerical directionality. When a predicted reward token is suboptimal, the policy gradient merely suppresses its probability mass without indicating the sign of the error—*i.e.*, whether the score should be increased or decreased to reach the ground truth. This non-directional optimization forces the model to explore the discrete token space blindly, making it difficult to provide clear guidance for distinguishing different quality levels.

Conversely, the Bradley-Terry (BT) loss models the reward as a continuous scalar. For a chosen and rejected pair $(\hat{\boldsymbol{q}}^w, \hat{\boldsymbol{q}}^l)$ with corresponding scores s_w and s_l, the loss is formulated as $\mathcal{L}_{\text{BT}} = -\log \sigma(s^w - s^l)$. The gradients with respect to the continuous rewards are:

$$\frac{\partial \mathcal{L}_{\text{BT}}}{\partial s^w} = -(1 - \sigma(s^w - s^l)), \quad \frac{\partial \mathcal{L}_{\text{BT}}}{\partial s^l} = 1 - \sigma(s^w - s^l) \tag{13}$$

Unlike policy gradients, the BT loss provides a deterministic, push-and-pull scalar force. The magnitude of this gradient is directly proportional to the current margin error $(1 - \sigma(\cdot))$. Furthermore, this formulation inherently acts as a built-in curriculum learning mechanism: it imposes substantial gradients on hard cases ($s^w \approx s^l$) and adaptively decays for easy cases ($s^w \gg s^l$), thereby ensuring highly efficient optimization.

B Analysis on Gradient Variance

Let π_θ denote an autoregressive MLLM policy Θ parameterized by θ. Given a multimodal input $\hat{\boldsymbol{q}} = (\boldsymbol{v}, \boldsymbol{c})$, the policy samples a token sequence $\boldsymbol{o}_i = (o_{i,1}, \ldots, o_{i,T_i})$, where $o_{i,t} \in \mathcal{V}$.

For the coupled generative paradigm, we consider a simplified policy-gradient form of GRPO. Given a group of G sampled responses $\{\boldsymbol{o}_i\}_{i=1}^G$ for the same input $\hat{\boldsymbol{q}}$, we focus on one sampled response and omit the sampling-chain index i for notational simplicity. The gradient contribution of a single sampling chain can then be written as

$$\hat{g}_{\text{GRPO}} \propto \hat{A} \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta \left(o_t \mid \hat{\boldsymbol{q}}, \boldsymbol{o}_{<t}\right), \tag{14}$$

where $\hat{A}$ denotes the group-normalized advantage of the sampled response $\boldsymbol{o}$. [11] empirically show that gradient variance increases with the length of the response trajectory. We further analyze this phenomenon from a theoretical perspective.

Theorem: GRPO Gradient Variance Scales as $\Omega(T)$

We begin by establishing the variance properties of the cumulative score function $\boldsymbol{S}$ under the standard autoregressive generation process, independent of any reward signal.

Lemma B.1. *Under the autoregressive generation process $\boldsymbol{o} \sim \pi_\theta$, let $\mathcal{H}_t = (\hat{\boldsymbol{q}}, o_1, \ldots, o_t)$ denote the trajectory history up to step t. The per-step score function $\boldsymbol{s}_t = \nabla_\theta \log \pi_\theta(o_t \mid \hat{\boldsymbol{q}}, \boldsymbol{o}_{<t})$ forms a Martingale Difference Sequence (MDS) conditioned on the history $\mathcal{H}_{t-1}$. Consequently, the variance of the cumulative score function $\boldsymbol{S} = \sum_{t=1}^{T} \boldsymbol{s}_t$ satisfies:*

$$\text{Var}[\boldsymbol{S}] \geq T \cdot G_0,$$

where $G_0 \triangleq \min_t \mathbb{E}[\|\boldsymbol{s}_t\|^2] > 0$ is the minimum per-step variance.

Proof. Specifically, the score function identity gives:

$$\begin{aligned}\mathbb{E}[\boldsymbol{s}_t \mid \mathcal{H}_{t-1}] &= \mathbb{E}_{o_t\sim\pi_\theta(\cdot\mid\hat{\boldsymbol{q}},\boldsymbol{o}_{<t})}\big[\nabla_\theta \log \pi_\theta(o_t \mid \hat{\boldsymbol{q}}, \boldsymbol{o}_{<t})\big] \\ &= \sum_{o_t} \pi_\theta(o_t \mid \hat{\boldsymbol{q}}, \boldsymbol{o}_{<t}) \nabla_\theta \log \pi_\theta(o_t \mid \hat{\boldsymbol{q}}, \boldsymbol{o}_{<t}) \\ &= \nabla_\theta \sum_{o_t} \pi_\theta(o_t \mid \hat{\boldsymbol{q}}, \boldsymbol{o}_{<t}) = \nabla_\theta 1 = 0. \end{aligned} \tag{15}$$

By the MDS property, cross-covariances vanish exactly:

$$\mathbb{E}[\langle \boldsymbol{s}_i, \boldsymbol{s}_j \rangle] = \mathbb{E}\left[\mathbb{E}[\langle \boldsymbol{s}_i, \boldsymbol{s}_j \rangle \mid \mathcal{H}_{j-1}]\right] = \mathbb{E}\left[\langle \boldsymbol{s}_i, \mathbb{E}[\boldsymbol{s}_j \mid \mathcal{H}_{j-1}] \rangle\right] = 0.$$

Therefore, the variance of $\boldsymbol{S}$ decomposes perfectly into the sum of per-step variances:

$$\mathrm{Var}[\boldsymbol{S}] = \mathrm{Var}\left[\sum_{t=1}^{T} \boldsymbol{s}_t\right] = \sum_{t=1}^{T} \mathrm{Var}[\boldsymbol{s}_t] \geq \sum_{t=1}^{T} G_0 = T \cdot G_0. \tag{16}$$

□

Theorem 1. *Under Lemma B.1, the variance of the GRPO gradient estimator satisfies*

$$\mathrm{Var}\big[\hat{g}_{\mathrm{GRPO}}\big] \;=\; \Omega(T).$$

Proof. We evaluate the variance of the coupled estimator, under the approximation that $\hat{A}$ and $\nabla_\theta \log \pi_\theta(o_t \mid \hat{\boldsymbol{q}}, \boldsymbol{o}_{<t})$ are independent, we have:

$$\mathrm{Var}\big[\hat{g}_{\mathrm{GRPO}}\big] = \mathrm{Var}[\hat{A} \cdot \boldsymbol{S}] \propto \mathrm{Var}[\boldsymbol{S}] \cdot \mathbb{E}[\hat{A}^2].$$

Finally, by construction of the group relative z-score normalization in GRPO, we have $\mathbb{E}[\hat{A}^2] = 1$. Therefore:

$$\mathrm{Var}\big[\hat{g}_{\mathrm{GRPO}}\big] \;\geq\; T \cdot G_0 \;=\; \Omega(T).$$

□

This theorem formally establishes that the gradient variance of the GRPO estimator scales with response length, directly causing the pronounced fluctuations observed in human preference accuracy during GRPO training. While the clipping mechanism in GRPO (i.e., truncating importance weights to $[1-\varepsilon, 1+\varepsilon]$) can partially control the gradient magnitude, it introduces additional optimization bias. This represents a bias-variance tradeoff rather than a fundamental resolution to the $\Omega(T)$ variance growth.

C Details on Data Collection

To construct our training dataset, we first caption collected real-world videos encompassing diverse subjects, dynamics, environments, styles, and camera movements. These captions are subsequently utilized as text prompts for video generation, effectively challenging T2V models to produce high-quality content that maintains strict semantic alignment with text prompt. We employ a suite of T2V models, including Gen-2 [32], Pika 1.0 [18], PixVerse (v1/v2) [27], Dreamina [4], Luma [1], Gen-3 [33], and Kling [17] to generate video pairs based on these prompts. Human annotators then evaluate these pairs, providing preference labels by analyzing alignment across five sub dimensions: object, dynamics, environment, style, and camera movement. The annotation team comprises 20 annotators and 14 reviewers. All annotators must achieve an accuracy above 90% on a qualification test. Each pair is labeled by one annotator and independently reviewed, with incorrectly labeled or ambiguous pairs excluded. An additional quality check is conducted on 10% of the pairs. Since each pair receives only one initial annotation, conventional inter-annotator agreement cannot be computed; instead, annotation reliability is supported by annotator qualification, independent review, and additional quality checks. In total, we collect 22K video pairs for training and an additional 1469 pairs as an in-domain dataset to benchmark DeScore. Annotators use five alignment dimensions as guidelines and provide one overall preference; no dimension-level labels are collected or aggregated. Tied or ambiguous pairs are discarded, so BT uses only strict winner–loser pairs.

The resulting preference data is then integrated into our two-stage training paradigm, where CoT data are strategically generated and filtered to match specific optimization objectives: (1) Discriminative Cold-Start. In this stage, the priority is to activate the scoring module rather than enforce reasoning quality. We employ Qwen3-VL-8B [2] to generate the CoTs. To ensure data reliability, we perform consistency-based filtering, retaining only pairs where the preference derived from the CoT aligns with ground-truth human labels. (2) Dual-Objective Reinforcement Learning (RL). To further refine reasoning quality via GRPO, we leverage Gemini-2.5-Pro [8] to produce high quality and fine-grained CoTs, including sub-dimension scores (covering subjects, dynamics, environment, style, and camera movement) alongside the final preference. Consistent with the first stage, these CoTs are selectively filtered based on preference accuracy to ensure high dataset fidelity.

D User Instruction

When evaluating human preference alignment across in-domain datasets and OOD benchmarks, we compare DeScore against state-of-the-art (SOTA) video reward models using their original prompts, which are designed for holistic video quality assessment. For DeScore, we apply the specific user instruction illustrated in Figure 5. To accurately assess whether the generated video faithfully reflects the semantics of the text prompt, our instruction guides the model through a structured reasoning process. Specifically, it first deduces the expected visual elements from the text prompt, and subsequently verifies their presence in the generated video, explicitly assigning scores within the CoT across five fine-grained sub-dimensions: subject, dynamics, camera, environment, and style. Finally, the generated CoT rationale is processed by the dedicated scoring module to output the final scalar reward, providing a robust quantitative evaluation of the video's alignment with the text.

User Instruction for Our DeScore

Role: You are a professional Video-Text Alignment Judge. Your job is to strictly evaluate whether a generated video matches the user's text prompt.

Inputs:

- **Text Prompt**: [Insert the text prompt here]
- **Video**: <video>

Step 1: Thinking Process (Chain of Thought)
You must perform your reasoning in plain text BEFORE generating the final `<answer>` block. Follow these three stages of thought:

1. **Prompt Decoding & Visual Expectation**: Analyze the text prompt to establish a "Gold Standard" for evaluation before looking at the video.
   - **Explicit Elements**: List specific entities, actions, texts, or objects mentioned.
   - **Implicit/Abstract Translation**: If the prompt contains abstract concepts (e.g., "loneliness", "cinematic", "chaos"), translate them into concrete visual indicators (e.g., "dark lighting", "shallow depth of field", "fast camera movement").
2. **Visual Evidence Extraction**: Look at the video and describe what you actually see objectively, without forcing a match to the prompt.
   - What/Who is the main subject?
   - What are the exact actions and movements?
   - Are there any visual artifacts, hallucinations, or typos in generated text?
3. **Dimensional Analysis & Score Calculation**: Compare the Visual Evidence against the Prompt Expectations across 5 dimensions.
   - **Subject (Primary)**: Accuracy of main entities and text generation (spelling).
   - **Dynamics (Primary)**: Accuracy of actions, physics, and temporal flow.
   - **Camera (Secondary)**: Movement, angles, shot types.
   - **Environment (Secondary)**: Background, setting, lighting.
   - **Style (Secondary)**: Art style, aesthetic, visual quality.

   Sub-Dimension Scoring: Assign **2** (Perfect), **1** (Partial/Minor flaws), **0** (Failure/Hallucination), or **N/A** (Not applicable).

Step 2: Final Output
After completing your thinking process, output the final scores strictly in the following JSON format enclosed within `<answer>` tags. Do not include any reasoning inside the JSON.

```
<answer>
{
  "dimensional_scores": {
    "subject": "0, 1, 2, or N/A",
    "dynamics": "0, 1, 2, or N/A",
    "camera": "0, 1, 2, or N/A",
    "environment": "0, 1, 2, or N/A",
    "style": "0, 1, 2, or N/A"
  },
  "overall_score": 1.00-5.00
}
</answer>
```

Figure 5: **User Instruction for Our DeScore during training and inference.**

E Detailed Implementation

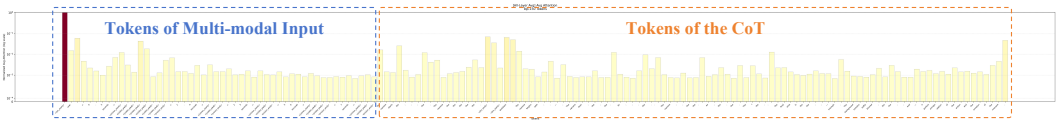
We employ Qwen3-VL-8B [2] as the backbone for DeScore. In the discriminative cold start stage, the model is fine-tuned on our constructed CoT dataset using LoRA with a rank of 64. The model is trained for two epochs using the AdamW optimizer with a learning rate of 2×10^{-6}, a weight decay of 0.01, and a batch size of 32. The resulting checkpoint is then used to initialize the subsequent stage. In the RL stage, we optimize a dual-objective comprising GRPO and BT losses. To balance the gradient scales between them, the coefficient for the BT-loss is set to 0.005, while the GRPO loss coefficient remains 1.0. We apply GRPO with a learning rate of 1×10^{-6} and a group size of $G = 8$. This stage is conducted for 65 steps with a rollout batch size of 128 and a mini-batch size of 32. For video processing during inference and training, the frame rate is set to 2 fps. All experiments are conducted on 8 GPUs.

F Detailed Evaluation Setting

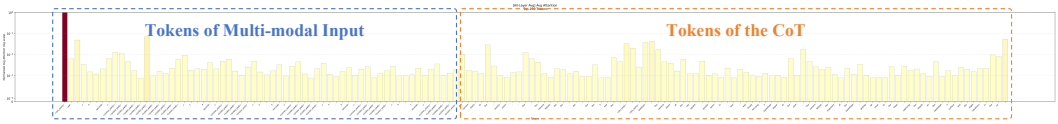
Baseline. To evaluate the capacity of DeScore in assessing diverse generative videos, we compare it against several video reward models. We first consider discriminative models, including VideoScore [9] and VideoAlign [21], which are trained to predict point-wise rewards using MSE loss and BT loss, respectively. Furthermore, we compare DeScore with state-of-the-art generative models, including VisionReward [48], UnifiedReward [45], UnifiedReward-Thinking [43], and VideoScore2 [9]. While the first two directly generate reward tokens, the latter two utilize a coupled sampling path that generates the CoT prior to the final reward to enhance interpretability. Specifically, UnifiedReward-Thinking is a pair-wise model trained to identify the superior video from a given pair.

Evaluation Benchmarks. Comparison experiments are conducted on an in-domain preference dataset consisting of 1,469 sample pairs. This dataset is a held-out subset partitioned from our original data source and remains unseen during training. To evaluate the generalization of DeScore, we further benchmark it on two Out-of-Distribution (OOD) suites: GenAI [15] and VideoGen-Bench [21, 55]. The former comprises 1.9k pairs generated by 10 early-stage T2V models across 508 prompts, typically characterized by lower resolutions ($\sim$320$\times$512) and shorter durations (2s-2.5s). In contrast, the latter contains 26.5k video pairs from 12 diverse open- and closed-source models across 420 prompts, featuring higher resolutions (up to 576$\times$1024), longer durations (4s–6s), and significantly improved visual quality. These two benchmarks represent early-generation and current state-of-the-art video models, respectively, providing a comprehensive assessment of DeScore across diverse generative scenarios.

Evaluation Metrics. Across diverse video reward benchmarks, we employ preference accuracy to assess the performance of DeScore, reporting both accuracy with ties (Acc w/ Tie) and without ties (Acc w/o Tie). Specifically, Acc w/o Tie directly compares the point-wise scores within each video pair, assigning the preference to the higher-scoring video. For Acc w/ Tie, if the absolute score difference between the two videos falls below a predefined threshold, the pair is classified as a tie. Furthermore, because the preference labels in both our training and evaluation datasets are fundamentally grounded in semantic fidelity to the text prompt, we specifically report the Text Alignment (TA) preference accuracy when evaluating DeScore on VideoGen-Bench. For state-of-the-art (SOTA) baselines, we utilize their corresponding task-specific scores for a fair comparison. Specifically, we evaluate VideoAlign using its dedicated text alignment (TA) sub-score. For UnifiedReward and VideoScore2, we explicitly apply their "text-to-video alignment" scores for



(a) Discriminative Cold Start Without Random Mask



(b) Discriminative Cold Start With Random Mask

Figure 6: **Visualization of the top 150 tokens most attended to by the `[Reward]` token.** Random masking encourages the model to attend more extensively to multi-modal input tokens, preventing the final reward prediction from relying solely on the CoT.

this dimension, while for all other remaining baseline models, we utilize their overall scores for performance comparison.

G Training Details of Ablation Study

Discriminative Version. This version shares a similar architecture with VideoAlign but employs Qwen3-VL-8B [2] as the backbone for feature extraction. Unlike DeScore, the learnable query token is appended directly after the multi-modal inputs without any explicit CoT generation. To ensure a fair comparison, it is trained on the exact same dataset and uses the identical configuration as the discriminative cold start stage of DeScore. Specifically, this discriminative baseline is fine-tuned via LoRA (rank of 64) for two epochs, using the AdamW optimizer with a learning rate of 2×10^{-6}, a weight decay of 0.01, and a batch size of 32.

Generative Version. This version follows the standard training paradigm for generative reward models, which first conducts supervised fine-tuning (SFT) and subsequently refines the model's capabilities via GRPO. It is formulated as a pair-wise model, given that only relative preferences, rather than absolute video reward scores, are accessible in the training dataset. The CoT data used for SFT is generated by Gemini-2.5-Pro following the approach detailed in Section 3.1. During the GRPO training stage, we incorporate additional data to ensure robust model convergence. The model is optimized using the AdamW optimizer with a weight decay of 0.01 and a learning rate of 1×10^{-6}. Specifically, the GRPO training utilizes a rollout group size of $G = 8$, a rollout batch size of 256, and an update mini-batch size of 64.

H Additional Experiments

H.1 Visualization

To validate our hypothesis that random masking encourages the model to jointly leverage multi-modal inputs and CoT, rather than over-relying on reasoning tokens, we visualize the top 150 tokens receiving the highest attention from the final reward query. As shown in Figure 6, random masking causes the query token to attend to both multi-modal inputs and CoT, rather than relying solely on reasoning tokens, which contributes to better reward performance.

H.2 Improving Video Generation

To further demonstrate the effectiveness of DeScore in improving generated video quality, we integrate it into two representative post-training paradigms, Longcat-GRPO [37] and Flow-DPO [21], on Wan-2.1-1.3B [39], and evaluate the resulting models on VBench [12]. We sample 10K prompts from those used to construct our preference video dataset. For Longcat-GRPO, we follow the official setting in its technical report. For Flow-DPO, we generate videos with Wan-2.1-1.3B and form positive and negative pairs according to reward scores from the corresponding reward model, with the pairs periodically updated during training following the official setting.

As shown in Table 3, DeScore consistently improves the quality of generated videos under both post-training paradigms. We further provide qualitative examples in Figure 7. Compared with the baseline Wan-2.1-1.3B, post-training with DeScore significantly improves video quality across multiple aspects, including subject fidelity, camera motion, spatial relationships, and dynamics. While the baseline model often fails to generate videos that faithfully match the text prompt, both Flow-DPO



Figure 7: **Qualitative examples of improved video generation with DeScore.**

Table 4: **Inference Efficiency and Latency Breakdown of DeScore and VideoScore2 under Identical Evaluation Settings.**

Method	Paradigm	Total latency (s/video)	Generation	Score readout	Gen. tokens	Throughput (video/s)	Decode speed (tok/s)
VideoScore2	generative CoT scoring	3.30 ± 0.49	3.30	– (in-text)	436	0.303	132.1
DeScore (ours)	CoT + reward head	3.94 ± 0.59	3.54	0.40	464	0.254	130.9

and Longcat-GRPO equipped with DeScore produce videos that better capture the semantic content of the prompt.

H.3 Inference Cost Analysis

To quantify the practical inference overhead introduced by DeScore's decoupled scoring mechanism, we compare it against VideoScore2, the most directly comparable CoT-based baseline, under identical evaluation settings. As shown in Tab. 4, both methods are evaluated on 100 videos using a single GPU with BF16 precision, a batch size of 1, and 2-FPS frame sampling. Video decoding and preprocessing are excluded from the latency measurements, as these operations are identical for both methods. After three warm-up runs for each method, we measure wall-clock latency with CUDA synchronization and use vLLM for CoT generation. VideoScore2 achieves an average total latency of 3.30 ± 0.49 seconds per video, whereas DeScore requires 3.94 ± 0.59 seconds, corresponding to a 19.4% increase in total latency. The resulting throughputs are 0.303 and 0.254 videos/s, respectively. Notably, DeScore spends 3.54 seconds per video on CoT generation, only 7.3% more than VideoScore2 (3.30 seconds). This modest increase is consistent with its slightly longer generated responses (464 vs. 436 tokens),

while maintaining a comparable token decoding speed (130.9 vs. 132.1 tok/s). The additional 0.40 seconds per video is attributable to a separate forward pass through the decoupled reward head to obtain the scalar reward, whereas VideoScore2 directly generates its score as part of the textual output. Overall, DeScore introduces a moderate inference overhead primarily due to its explicit reward-readout stage, while maintaining a CoT decoding efficiency comparable to VideoScore2. We will include the complete evaluation setup and the corresponding latency, throughput, generated-token, and decoding-speed measurements in the revised manuscript.

I Limitation and Future Work

Although DeScore demonstrates strong generalization across diverse scenarios, it is primarily evaluates whether generated videos are faithful to the input text prompt. Consequently, it may be less effective at capturing motion implausibility and visual artifacts. In future work, we plan to extend our decoupled paradigm to multi-dimensional video reward modeling, aiming to build a more comprehensive reward model for advancing video generation.

J Ethics statement

Our work focuses on algorithmic improvements for video reward modeling, with the goal of more accurately assessing the quality of generated videos. The training data consist of videos generated by open-source and closed-source generative models, and the evaluation is conducted on publicly available benchmarks, including GenAI, VideoGen-Bench, and VBench. No data are collected from human subjects. We do not anticipate any direct, immediate, or negative societal impacts arising from this research.

K Reproducibility statement

To ensure reproducibility, we provide detailed descriptions of datasets, training configurations, and hyper-parameters in both the main text and supplementary materials. We will provide our code to facilitate reproducibility.

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 clearly state the claims made, including the contributions made in the paper and important assumptions and limitations.

   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: In Appendix I.

   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: [Yes]

Justification: In Appendix B.

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: In Appendix K.

   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: In Appendix K.

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: In Appendix E.

   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: We follow the widely-used evaluation benchmark, and these metrics do not require reporting error bars.

   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: All experiments are conducted on 8 GPUs as shown in Appendix E.

   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: [Yes]

   Justification: We have reviewed and followed the NeurIPS Code of Ethics.

   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: [Yes]

    Justification: In Appendix J.

    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: [Yes]

    Justification: This paper poses no such risks.

    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: We have carefully cited and stated the assets used in the paper.

    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 release new assets at submission time because the code, checkpoints, and processed training data are currently under institutional review. Appendix E and Appendix C documents the data sources and experimental setup, and we will release the assets together with usage instructions and documentation once the review process is completed.

    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 nor research 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 crowdsourcing nor research with human subjects.

    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 rigorousness, 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.