

RAVEN: Real-time Autoregressive Video Extrapolation with Consistency-model GRPO

Yanzuo Lu **Ronglai Zuo** **Jiankang Deng**
MVP Lab, Imperial College London
https://yanzuo.lu/raven

Abstract

Causal autoregressive video diffusion models support real-time streaming generation by extrapolating future chunks from previously generated content. Distilling such generators from high-fidelity bidirectional teachers yields competitive few-step models, yet a persistent gap between the history distributions encountered during training and those arising at inference constrains generation quality over long horizons. We introduce the Real-time Autoregressive Video Extrapolation Network (RAVEN), a training-time test framework that repacks each self rollout into an interleaved sequence of clean historical endpoints and noisy denoising states. This formulation aligns training attention with inference-time extrapolation and allows downstream chunk losses to supervise the history representations on which future predictions depend. We further propose Consistency-model Group Relative Policy Optimization (CM-GRPO), which reformulates a consistency sampling step as a conditional Gaussian transition and applies online Reinforcement Learning (RL) directly to this kernel, avoiding the Euler-Maruyama auxiliary process adopted in prior flow-model RL formulations. Experiments demonstrate that RAVEN surpasses recent causal video distillation baselines across quality, semantic, and dynamic degree evaluations, and that CM-GRPO provides further gains when combined with RAVEN.

1 Introduction

Recent progress in video diffusion has established bidirectional models as the dominant paradigm for high-fidelity generation [1–15]. Their reliance on bidirectional context and a large number of denoising steps, however, limits their suitability for real-time generation, where video must be produced continuously as a stream. This requirement has motivated causal autoregressive architectures that extrapolate future chunks from previously generated content [16–30]. The strongest generation capability still largely resides in high-step bidirectional models, and recent work has studied asymmetric distillation, which transfers knowledge from such bidirectional teachers to causal student generators [31–36]. The resulting few-step generators achieve real-time generation speeds while retaining much of the visual fidelity of their teachers.

A central challenge in autoregressive video diffusion distillation lies in how the model represents and reuses historical chunks, as each generated chunk becomes the context on which all subsequent predictions depend. As illustrated in Figure 1, existing training paradigms differ in both the source of historical states and whether those states receive end-to-end supervision from later chunks. Teacher Forcing trains with real historical chunks, which provides clean supervision but does not expose the generator to its own test-time history. Diffusion Forcing [37, 38] trains causal diffusion models by assigning each token an independently sampled Signal-to-Noise Ratio (SNR), and CausVid [35] adapts this construction to autoregressive video distillation by incorporating Distribution Matching Distillation (DMD) [39, 40]. This formulation optimizes the causal generator under a history

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

distribution that does not match inference, and the resulting discrepancy can accumulate across autoregressive rollouts. Self Forcing [31] reduces this discrepancy by conditioning the DMD objective on self rollouts, yet the historical cache is reused as detached context, so the history representations receive no end-to-end supervision from subsequent chunk losses.

We propose the Real-time Autoregressive Video Extrapolation Network (RAVEN), a training-time test framework that directly supervises the history construction used during autoregressive extrapolation. Starting from self rollouts of the few-step causal generator, RAVEN repacks the sampled trajectory into an interleaved sequence of clean historical endpoints and noisy denoising states. Within this sequence, clean rollout chunks provide the causal history for subsequent predictions, while noisy states from the same rollout remain the supervised denoising inputs. The resulting attention computation aligns more closely with inference than Teacher Forcing or Diffusion Forcing and keeps history representations inside the supervised forward pass, as shown in Figure 1(d). This design enables gradients from later chunks to shape the cached representations on which future predictions depend, while avoiding the cost of backpropagating through an entire autoregressive sampling trajectory.

Reinforcement learning (RL) has become an influential post-training paradigm for large generative models, and recent work has begun to adapt it to diffusion and flow models. Flow-GRPO [41] demonstrates this direction for flow matching, addressing the conflict between deterministic Ordinary Differential Equation (ODE) sampling and the stochastic exploration required by policy optimization through an ODE-to-Stochastic Differential Equation (SDE) conversion followed by Euler-Maruyama discretization. The causal generator in RAVEN employs a few-step consistency sampler, for which Euler-Maruyama introduces a train-test discrepancy by optimizing over stochastic transitions that differ from the deterministic sampling used at inference. We observe that a consistency sampling step can be cast as a conditional Gaussian transition parameterized by the predicted clean endpoint, enabling the policy objective to be defined on the same update rule used during generation without an auxiliary stochastic process. This correspondence is especially consequential for autoregressive video generation, where each generated chunk alters the history on which subsequent predictions depend. We therefore propose Consistency-model Group Relative Policy Optimization (CM-GRPO), which applies group relative policy optimization directly to this consistency transition kernel.

Our contributions are as follows.

- We identify a history supervision gap in autoregressive video diffusion distillation, where existing methods are either optimized under history distributions that differ from inference or conditioned on rollout history without end-to-end supervision.
- We introduce RAVEN, a training-time test framework that repacks self rollouts into an interleaved sequence of clean historical endpoints and noisy denoising states, allowing supervision to propagate through the history representations used during extrapolation.
- We propose CM-GRPO, which reformulates a consistency sampling step as a conditional Gaussian transition kernel and applies group relative policy optimization directly to this kernel, matching the sampler interface used at inference.
- We demonstrate that RAVEN surpasses recent causal video distillation baselines and that CM-GRPO provides complementary gains when combined with RAVEN.

2 Related Work

Autoregressive Video Diffusion Distillation. Autoregressive video generation encompasses several parallel directions beyond the causal distillation setting studied in this paper. One line of work explores the design of the autoregressive rollout itself, either extending the prediction window for longer sequences or conditioning on intermediate noisy latents rather than fully denoised outputs as historical context [42, 43, 32, 44, 34, 45]. Although our current implementation conditions on clean latents, the training-time test paradigm can simulate these alternative history mechanisms to provide end-to-end supervision. A separate direction develops architectures with dedicated temporal memory for managing long-range context during training [42, 46–51], while a complementary body of training-free methods adapts models at inference time for length extrapolation [52–57]. Our framework is orthogonal to both families, as any strategy that generates and caches the next chunk through specialized memory designs can be executed within the self-rollout phase and benefit from the subsequent interleaved optimization.

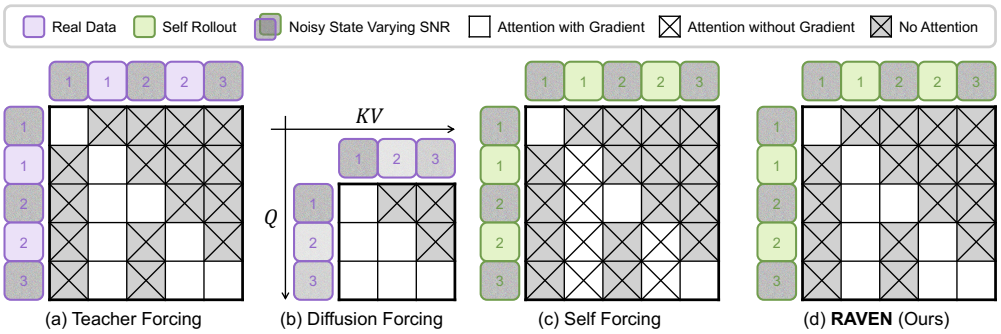


Figure 1: **Attention Mask Configuration.** Autoregressive video diffusion training paradigms differ in how historical states enter attention and whether those states receive end-to-end supervision from later chunks. Teacher Forcing and Diffusion Forcing rely on data-driven historical states, inducing a training distribution that differs from inference. Self Forcing shifts the history distribution toward inference but reuses the historical cache as detached context. RAVEN instead repacks each self rollout into clean historical endpoints and noisy denoising states, allowing later chunks to attend to the same history used during extrapolation while their losses supervise the cached representations.

Online RL in Diffusion Model. Online RL has become a practical paradigm for aligning diffusion and flow models after pretraining, beginning with reward-guided optimization for image generation and gradually evolving into policy optimization methods tailored to diffusion and flow trajectories [58, 41, 59–61]. This approach has since been extended to autoregressive generators and world models, where reinforcement learning serves not only for preference alignment but also for preserving pretrained capabilities and improving controllable generation over long horizons [62–67]. Parallel work applies online RL to distilled and few-step generators, where the central challenge is to improve alignment without sacrificing the efficiency that makes these models practical [68–71]. Much of the follow-up work has focused on refining the policy objective itself. Some methods revisit regularization to control reward hacking and distribution drift [72–75], while others study how the stochasticity or numerical form of the sampler shapes policy optimization [76–83]. A separate direction makes more deliberate use of the denoising trajectory, for instance through branching, tree search, or stepwise credit assignment [84–100]. Our method is most closely related to the literature on few-step generation and sampler design. Rather than adopting the Euler-Maruyama discretization used in prior online RL formulations for flow models, CM-GRPO formulates the policy objective directly on the consistency transition kernel and combines it with the training-time test framework of RAVEN, more closely matching the inference-time behavior of autoregressive video extrapolation.

3 Methodology

3.1 Preliminaries

Let $x_{1:T}$ denote a sequence of latent video chunks and c the text condition, with hats used for student-generated quantities. Throughout the paper, the subscript t indexes the chunk position, while a superscript in parentheses, such as (n), (u), or (s), denotes the noise level. We write the autoregressive video diffusion model as

$$p_\theta(x_{1:T} \mid c) = \prod_{t=1}^{T} p_\theta(x_t \mid h_t, c), \qquad h_t = \mathcal{H}(x_{<t}). \tag{1}$$

The operator $\mathcal{H}(\cdot)$ denotes the history representation encoded by the model via its cache. For a noise level n, we define the noisy current chunk as $z_t^{(n)} = \alpha_n x_t + \sigma_n \epsilon$, with $\epsilon \sim \mathcal{N}(0, I)$. Training paradigms are distinguished primarily by how the history h_t is constructed from past chunks, and we detail this distinction in the following subsections.

History Formulation in Diffusion Forcing and Self Forcing. Recent methods for autoregressive video diffusion distillation are largely built on either Diffusion Forcing [37] or Self Forcing [31]. In CausVid [35], training follows Diffusion Forcing and represents the history as $h_t^{\mathrm{DF}} = \mathcal{H}\big(z_1^{(n_1)}, \ldots, z_{t-1}^{(n_{t-1})}\big)$, perturbing each ground-truth prefix chunk with an independently sam-

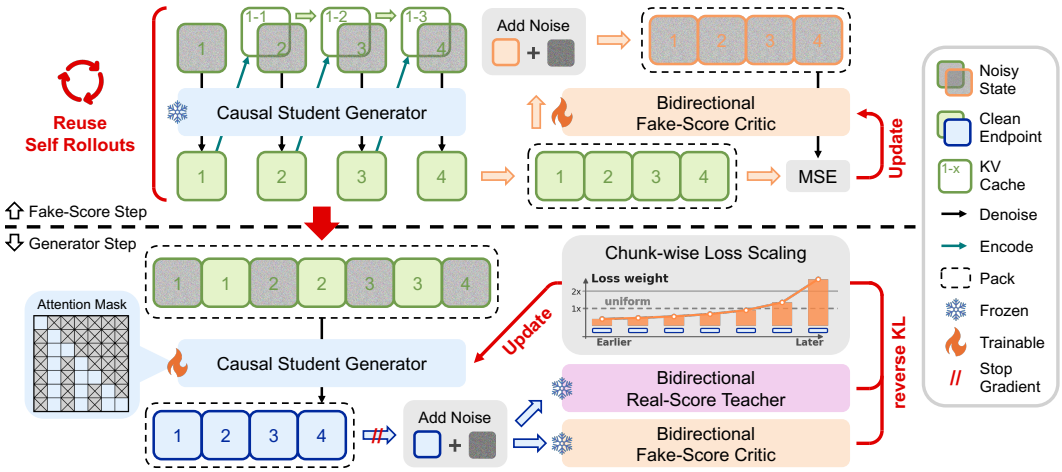


Figure 2: **Training Pipeline.** RAVEN builds on score distillation with a training-time test formulation that aligns the generator's training context with inference. In the fake-score step, the frozen generator performs autoregressive self rollout with KV cache reuse, producing the clean endpoints and noisy denoising states that are subsequently reused in the generator step. Rather than discarding these rollout states after critic training, RAVEN repacks them into an interleaved sequence of clean historical endpoints and noisy denoising states, processed under a causal attention mask so that each noisy state attends to the clean history the generator itself produced. This allows later chunk losses, scaled chunk-wise, to supervise the history representations on which future predictions depend.

pled noise level before entering the causal context. Self Forcing [31] instead unrolls the autoregressive generator at training time and reuses detached cache representations written as $h_t^{\mathrm{SF}} = \mathrm{sg}(\mathcal{H}(\hat{x}_{<t}))$, where the stop-gradient operator $\mathrm{sg}(\cdot)$ treats historical chunks as fixed context for subsequent denoising steps. Both formulations therefore leave the cache construction outside end-to-end supervision, motivating the training-time test formulation introduced next.

Euler-Maruyama Discretization in Flow-GRPO. Flow-GRPO [41] starts from the rectified-flow ODE $\mathrm{d}y_\tau = v_\theta(y_\tau, \tau, c)\,\mathrm{d}\tau$, where y_τ denotes the latent variable at denoising time $\tau \in [0, 1]$. To inject the stochasticity required for policy optimization, it introduces an ODE-to-SDE conversion and operates on the reverse-time SDE $\mathrm{d}y_\tau = b_\theta(y_\tau, \tau, c)\,\mathrm{d}\tau + \sigma_\tau\,\mathrm{d}w$, where $b_\theta(y_\tau, \tau, c)$ is the drift term and $\sigma_\tau\,\mathrm{d}w$ the diffusion term. The drift term is given by

$$b_\theta(y_\tau, \tau, c) = v_\theta(y_\tau, \tau, c) + \frac{\sigma_\tau^2}{2\tau}\big(y_\tau + (1-\tau)v_\theta(y_\tau, \tau, c)\big). \tag{2}$$

Applying Euler-Maruyama discretization yields

$$y_{\tau+\Delta\tau} = y_\tau + b_\theta(y_\tau, \tau, c)\Delta\tau + \sigma_\tau\sqrt{\Delta\tau}\,\epsilon, \quad \epsilon \sim \mathcal{N}(0, I). \tag{3}$$

Equivalently, the Euler-Maruyama step defines an isotropic Gaussian policy kernel,

$$\pi_\theta^{\mathrm{EM}}\big(y_{\tau+\Delta\tau} \mid y_\tau, c\big) = \mathcal{N}\big(y_{\tau+\Delta\tau}; y_\tau + b_\theta(y_\tau, \tau, c)\Delta\tau, \sigma_\tau^2\Delta\tau I\big)\,. \tag{4}$$

This auxiliary kernel makes the policy ratio and the KL term tractable in closed form, but its stochastic transitions remain absent from the deterministic ODE sampler used at inference. ODE-based samplers are typically deterministic [101–107], while the consistency sampler [108–112, 40, 113–115] is a notable exception in the few-step regime, remaining defined on the probability flow ODE trajectory while still yielding stochastic transitions that can serve as the policy interface directly.

3.2 Training-Time Test via RAVEN

RAVEN is a training-time test framework for autoregressive video diffusion that aligns the training procedure with inference-time extrapolation. Building upon the asymmetric distillation formulated by CausVid [35], the pipeline distills knowledge from a frozen bidirectional teacher into the causal student generator. As illustrated in Figure 2, training alternates between a fake-score step and a generator step. In the fake-score step, the bidirectional fake-score critic is updated on self-rollout

samples perturbed with Gaussian noise. In the generator step, the causal student generator is updated via a reverse Kullback-Leibler (KL) score gradient computed from evaluations by both the bidirectional real-score teacher and the learned fake-score critic.

Let $\tau_1 > \cdots > \tau_K = 0$ denote the few-step sampling timesteps of the consistency sampler adopted by the generator. During the fake-score step, the frozen causal student generator autoregressively produces, for each chunk index t, a full denoising trajectory $\{\hat{z}_t^{(\tau_k)}\}_{k=1}^K$ along with the clean endpoint $\hat{x}_t = \hat{z}_t^{(0)}$. These clean endpoints are perturbed with Gaussian noise to form the training inputs for the fake-score critic. During the generator step, the same self rollout is reused and the noisy state at denoising level τ_k is taken directly from each chunk's sampled trajectory. These rollout states are then packed into an input sequence processed under the attention mask illustrated in Figure 1(d). Specifically, for a sampled timestep $u \in \{\tau_1, \ldots, \tau_{K-1}\}$, the interleaved sequence takes the form

$$\mathcal{I}_u = \big(\hat{z}_1^{(u)}, \hat{x}_1, \hat{z}_2^{(u)}, \hat{x}_2, \ldots, \hat{z}_{T-1}^{(u)}, \hat{x}_{T-1}, \hat{z}_T^{(u)}\big), \tag{5}$$

where $\hat{z}_t^{(u)}$ is the noisy state of chunk t at denoising level u and $\hat{x}_t$ is the corresponding clean endpoint. Within this sequence, the noisy states serve as supervised denoising targets, while the clean endpoints preceding chunk t constitute its history $h_t^{\mathrm{RAVEN}} = \mathcal{H}(\hat{x}_{<t})$. The causal student generator encodes these clean endpoints as history representations within the same forward pass, allowing later noisy states to attend to them under the causal attention structure employed during autoregressive extrapolation. The resulting predictions are subsequently perturbed with Gaussian noise and evaluated by the bidirectional real-score teacher and the fake-score critic to compute the reverse KL score gradient.

Reuse Self Rollouts. The formulation is inspired by the training-time test principle of EAGLE-3 [116], where the model is trained on the context it will produce and encounter during speculative decoding. In language generation, this amounts to feeding a predicted draft token representation into the next simulated drafting step. The analogous construction is substantially more involved for autoregressive video diffusion, since each chunk is the endpoint of a multi-step denoising trajectory and future chunks depend on the resulting cache. A direct simulation would require unrolling the generator across all chunks and denoising steps within a single computation graph, incurring backpropagation through both autoregressive recursion and sampler dynamics. RAVEN avoids this cost by exploiting the self rollout already produced during the fake-score step, which is precisely the process that defines future context at inference. Repacking its states into an interleaved sequence, where generated clean chunks supply context and later noisy states remain supervised targets, reduces training-time test to a reorganization of existing self rollouts rather than an additional mechanism layered on top of score distillation, while faithfully preserving the dependency structure of autoregressive extrapolation.

Chunk-wise Loss Scaling. Within the interleaved training sequence, chunks along the autoregressive horizon are exposed to qualitatively different denoising conditions. Earlier chunks operate under limited historical context, whereas later chunks condition on richer accumulated history and must simultaneously maintain contextual consistency and suppress error propagation. To account for this positional asymmetry, we introduce a future participation score. For a sequence of J chunks, let m_j denote the number of scalar elements in chunk j and let ℓ_j denote its summed loss. The future participation score is defined as $p_j = \sum_{k=j}^{J} m_k \,/\, \sum_{k=1}^{J} m_k$, namely the fraction of supervised elements contributed by chunk j and all subsequent chunks, which is larger for earlier chunks and decreases monotonically toward later ones. The resulting profile $\mathcal{P} = (p_1, \ldots, p_J)$ is passed to a predefined weighting function g_η to produce nonnegative raw weights $\tilde{w}_{1:J} = g_\eta(\mathcal{P})$, whose specific form is examined in the ablation studies. For any choice of g_η, the normalized per-chunk weights and the aggregate chunk loss are given by $w_j = \tilde{w}_j \sum_{k=1}^{J} m_k \,/\, \sum_{k=1}^{J} \tilde{w}_k m_k$ and $\mathcal{L}_{\mathrm{chunk}} = \sum_{j=1}^{J} w_j \ell_j \,/\, \sum_{j=1}^{J} w_j m_j$. The normalization ensures that the average element-wise weight is preserved, so g_η governs only the relative distribution of gradient emphasis across chunk positions. The complete training procedure is summarized in Algorithm 1 of Appendix A.

3.3 Online RL via CM-GRPO

CM-GRPO is an online policy optimization method for few-step consistency generators. As discussed in the preliminaries, Flow-GRPO [41] achieves tractable policy optimization for flow matching by converting the deterministic ODE into an auxiliary SDE via Euler-Maruyama discretization, yet the

resulting stochastic transitions are absent from the ODE sampler used at inference. A consistency sampler, by contrast, inherently yields stochastic Gaussian transitions through its predicted clean endpoint, enabling CM-GRPO to formulate the policy objective directly on the consistency transition kernel without introducing any auxiliary stochastic process.

Consider a single consistency sampling step from noise level u to a lower level s. Given the current latent $\tilde{z}^{(u)}$ and condition c, the model predicts a clean endpoint $\hat{x}_\theta = f_\theta(\tilde{z}^{(u)}, u, c)$, from which the next latent is drawn as $\tilde{z}^{(s)} = \alpha_s \hat{x}_\theta + \sigma_s \epsilon$ with $\epsilon \sim \mathcal{N}(0, I)$, where α_s and σ_s are the noise schedule coefficients. This sampling rule induces the Gaussian transition probability

$$\pi_\theta\big(\tilde{z}^{(s)} \mid \tilde{z}^{(u)}, c\big) = \mathcal{N}\Big(\tilde{z}^{(s)}; \mu_\theta^{u\to s}, \sigma_s^2 I\Big), \qquad \mu_\theta^{u\to s} = \alpha_s \hat{x}_\theta, \tag{6}$$

which constitutes the policy interface in CM-GRPO.

To instantiate group relative policy optimization on this kernel, for each condition c the generator runs G independent consistency trajectories, each terminating in a clean output $\hat{x}^i$ on which a scalar reward R_i is evaluated. Following GRPO [117], the group-normalized advantage is computed as $\hat{A}_i = \big(R_i - \mathrm{mean}(\{R_j\}_{j=1}^G)\big) \,/\, \big(\mathrm{std}(\{R_j\}_{j=1}^G) + \epsilon\big)$. This advantage is broadcast to all consistency sampling transitions within the same trajectory, converting the endpoint reward into a per-transition objective. For a transition from u to s, dropping the Gaussian normalization constant and terms independent of θ, the log probability under the consistency kernel reduces to

$$\log \pi_\theta\big(\tilde{z}_i^{(s)} \mid \tilde{z}_i^{(u)}, c_i\big) = -\frac{\|\tilde{z}_i^{(s)} - \mu_\theta^{u\to s}\|^2}{2\sigma_s^2}. \tag{7}$$

Because $\mu_\theta^{u\to s} = \alpha_s \hat{x}_\theta$, the gradient of the advantage-weighted log probability with respect to the predicted clean endpoint takes the form

$$\nabla_{\hat{x}_\theta}\Big[-\hat{A}_i \log \pi_\theta(\tilde{z}_i^{(s)} \mid \tilde{z}_i^{(u)}, c_i)\Big] = -\hat{A}_i \alpha_s \frac{\tilde{z}_i^{(s)} - \mu_\theta^{u\to s}}{\sigma_s^2}. \tag{8}$$

CM-GRPO implements this update through the stop-gradient regression objective

$$\mathcal{L}_{\text{CM-GRPO}} = \mathbb{E}_{i,u,s}\left[\left\|\hat{x}_\theta - \mathrm{sg}\left(\hat{x}_\theta + \frac{\hat{A}_i \alpha_s}{2\sigma_s^2}\big(\tilde{z}_i^{(s)} - \mu_\theta^{u\to s}\big)\right)\right\|^2\right], \tag{9}$$

whose gradient with respect to $\hat{x}_\theta$ recovers exactly the endpoint gradient derived above, matching the score gradient update used in our implementation. The same formulation also admits reference policy KL regularization. If a reference consistency model produces a clean endpoint $\hat{x}_{\mathrm{ref}}$ under the same noisy state $\tilde{z}^{(u)}$, the KL divergence between the two Gaussian kernels reduces to

$$D_{\mathrm{KL}}\Big(\pi_\theta(\cdot \mid \tilde{z}^{(u)}, c)\Big\|\pi_{\mathrm{ref}}(\cdot \mid \tilde{z}^{(u)}, c)\Big) = \frac{\alpha_s^2 \|\hat{x}_\theta - \hat{x}_{\mathrm{ref}}\|^2}{2\sigma_s^2}. \tag{10}$$

This regularizer is tractable in principle, but in our current implementation the bidirectional teacher cannot be sampled through the consistency interface and therefore does not provide $\hat{x}_{\mathrm{ref}}$ on this policy interface. We therefore derive this closed-form expression for completeness, leaving its practical application to future work in which a compatible reference consistency model is accessible. The complete training procedure is summarized in Algorithm 2 of Appendix A.

Reward Composition. Autoregressive video reinforcement learning requires reward signals that jointly capture motion dynamics, visual fidelity, and semantic alignment. We empirically find that overweighting visual fidelity or semantic alignment tends to encourage static generations, whereas an overly strong motion reward degrades the remaining two aspects, making reward design challenging. This difficulty is compounded by the limited availability of reliable holistic metrics for few-step video generation. Reward models built on vision-language models (VLMs) [118, 119] or diffusion latents [120] supply useful scalar preferences, yet their preference data are typically collected from high-step or high-quality generators, introducing a distribution shift when applied to outputs of few-step distilled models. We therefore combine VLM-based rewards with rewards derived from representation models [121–124]. Each reward component is normalized within the sampled group before weighted summation, balancing reward scales and preventing any single metric from dominating the group-relative advantage.



Figure 3: Qualitative comparisons. Playable video clips are available on our project page.

4 Experiments

Implementation. All experiments are built upon Wan2.1-T2V-1.3B [13] as the base model with 3 latent frames per chunk, consistent with existing baselines. RAVEN adopts the same initialization as Causal Forcing [36], which grounds the causal student in ODE distillation from an autoregressive teacher to satisfy frame-level injectivity. The CM-GRPO stage then proceeds from the RAVEN checkpoint. For reward composition, we integrate several representation models derived from VBench [125] consisting of complementary dimensions with a VLM-based reward trained on human feedback for *Text Alignment*. The representation rewards span both temporal dynamics (i.e., *Dynamic Degree* and *Motion Smoothness*) and frame-level visual quality (i.e., *Aesthetic Quality* and *Imaging Quality*). More implementation details can be found at Appendix B.

Evaluation. We adopt VBench [125] and report the *Total Score* along with the *Quality Score* and *Semantic Score*, reproducing every baseline from its released checkpoint and inference code under the same environment, prompts, seeds, and VBench version. Because the *Dynamic Degree* dimension in VBench is computed from RAFT [124] optical-flow magnitudes and therefore picks up camera jitter and temporal drift alongside genuine motion, we additionally adopt UnifiedReward-32B [119] to evaluate dynamic degree on the full collection of 6,220 videos generated under the VBench prompt suite. User studies are reported in Appendix C and seed uncertainty in Appendix B.

4.1 Comparison with Baselines

Quantitative Comparisons. As reported in Table 1, earlier causal distillation baselines cluster within a narrow range on the total score, and gains on quality or semantic alignment are typically offset by reduced motion or the reverse. Causal Forcing [36] pushes the overall scores higher and partially recovers motion among the baselines, yet the same trade-off persists. RAVEN surpasses every prior baseline across all four

Table 1: Quantitative comparison results.

Method	Total Score	Qual. Score	Sem. Score	Dyn. Deg.
CausVid [35]	83.01	84.18	78.34	2.340
LongLive [34]	83.05	83.70	80.46	2.277
Rolling Forcing [32]	83.25	84.00	80.25	2.536
Self Forcing [31]	84.27	85.10	80.97	2.543
Reward Forcing [33]	84.39	85.27	80.87	2.508
Causal Forcing [36]	84.96	86.00	80.76	2.669
+ CM-GRPO	85.08	86.12	80.96	2.829
RAVEN	85.15	86.18	81.04	2.951
+ CM-GRPO	**85.46**	**86.54**	**81.17**	**2.962**

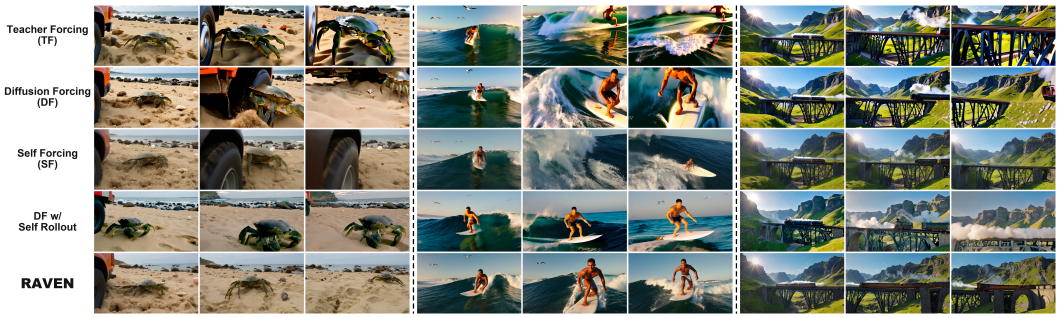


Figure 4: Qualitative ablation on Training-time Test. Playable video clips are available on our project page.

dimensions, with the largest margin on dynamic degree, indicating that supervising the cached history alleviates this trade-off rather than redistributing it. Adding CM-GRPO to Causal Forcing yields a smaller gain concentrated on motion, whereas its combination with RAVEN produces the leading entry on every dimension, suggesting that the two contributions are complementary and that the policy update benefits from a generator already aligned with inference.

Qualitative Comparisons. Figure 3 contrasts Causal Forcing and RAVEN, each with and without CM-GRPO, across four prompts spanning animal motion, urban scenery, and human subjects. Causal Forcing exhibits severe structural failures under motion, stretching the corgi's body into an unnaturally long shape, detaching the lion's head from its body, and rendering colors in an over-saturated tone that looks unnatural. Adding CM-GRPO repairs these structural breakdowns and the facial distortion of the running child, while RAVEN avoids the same failures from the start and produces more realistic colors and proportions, with only mild blurring around fast-moving regions such as the boy's face and the lion's body. Combining RAVEN with CM-GRPO yields the most coherent results, sustaining structural stability and temporal smoothness through continuous motion such as the woman's turning hair and the child running in the rain.

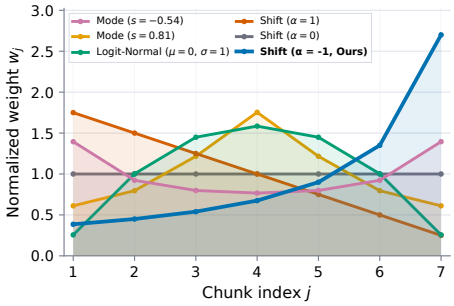
Table 2: Ablation on Training-time Test.

Method	Total Score	Qual. Score	Sem. Score	Dyn. Deg.
Teacher Forcing (TF)	82.64	83.11	80.73	**3.000**
Diffusion Forcing (DF)	84.09	84.75	81.45	2.743
Self Forcing (SF)	84.06	84.68	**81.56**	2.347
DF w/ Self Rollout	83.30	83.96	80.65	2.979
RAVEN	**85.15**	**86.18**	81.04	2.951

4.2 Ablation Studies

Effect of Training-time Test. All entries in Table 2 share the same ODE-distilled initialization and chunk-wise loss scaling, isolating the effect of how history is formed and supervised. TF achieves the strongest motion at the cost of the other two dimensions, while SF leads on semantic alignment but yields the weakest motion, reflecting how a detached cache withholds gradient from the history. Replacing real prefixes in DF with self-rollout endpoints recovers motion close to TF yet erodes quality and semantic alignment, indicating that aligning the history distribution without supervising it merely redistributes the error. RAVEN routes gradients from later chunks through the clean rollout endpoints used as history, attaining the leading total score while retaining motion close to TF. These quantitative tradeoffs are visually validated in Figure 4, where RAVEN consistently maintains structural coherence and object identity across frames, avoiding the severe structural distortions of TF and the blurring artifacts of SF.

Effect of Chunk-wise Loss Scaling. We compare candidate weighting functions g_η that map the future participation score p_j to a per-chunk weight, summarized in Figure 5. Two candidates are inherited from SD3 [126], the mode density at $s \in \{-0.54, 0.81\}$ and the logit-normal density at $(\mu, \sigma) = (0, 1)$, each integrated over the participation interval covered by chunk j to produce its raw weight. The remaining family follows the shift parameterization $\pi_\alpha(p) = \alpha p/(1 + (\alpha - 1)p)$ borrowed from flow-matching timestep schedules, where $\alpha = 1$ biases mass toward early chunks and $\alpha = 0$ recovers a uniform schedule, while a negative α applies $\pi_{|\alpha|}$ to the reversed coordinate p_J/p_j so that emphasis concentrates on later chunks instead. Profiles peaked near the middle of the rollout or biased toward early chunks fall below the uniform baseline on total and quality, with the shift at $\alpha = 1$ trading semantic alignment for the highest dynamic degree. The shift at $\alpha = -1$, the setting we adopt throughout RAVEN, instead raises the total score by roughly 1.3 points while keeping motion and semantic alignment competitive. This gain does not come from the weighting schedule



Weighting Function g_η	Total Score	Qual. Score	Sem. Score	Dyn. Deg.
Mode ($s = -0.54$)	82.58	82.86	**81.43**	2.996
Mode ($s = 0.81$)	83.08	83.74	80.47	2.971
Logit-Normal ($\mu = 0$, $\sigma = 1$)	83.31	83.97	80.65	2.963
Shift ($\alpha = 1$)	83.79	84.87	79.46	**3.000**
Shift ($\alpha = 0$)	83.82	84.67	80.42	2.924
Shift ($\alpha = -1$, Ours)	**85.15**	**86.18**	81.04	2.951

Figure 5: Ablation on Chunk-wise Loss Scaling.

alone. The Self Forcing entry in Table 2 applies the same schedule to the detached paradigm and falls below the uniformly weighted Causal Forcing in Table 1 on both total score and dynamic degree. Later chunks accumulate more drift and attend to the longest self-generated prefix, so upweighting them in RAVEN improves their own denoising and at the same time sends gradient into the cached history they depend on. In a detached baseline the history is a fixed input, and the same upweighting only asks the generator to denoise better under a context it cannot improve.

Effect of Reward Composition. Table 3 ablates each reward dimension around our adopted composition. Halving the *Text Alignment* (TA) weight lowers the total and quality scores without a semantic gain, and reducing the *Dynamic Degree* (DD) weight weakens both the total score and motion, indicating that the optical flow term carries the bulk of motion supervision rather than acting as a redundant complement to *Motion Smoothness* (MS). Raising the MS weight produces a smaller motion regression, while doubling *Aesthetic Quality* (AQ) and *Imaging Quality* (IQ) lifts semantic alignment to the highest entry at the cost of motion. The adopted composition therefore best balances the three aspects identified in the methodology.

Table 3: Ablation on Reward Composition.

Reward Dimensions					Total Score	Qual. Score	Sem. Score	Dyn. Deg.
TA	DD	MS	AQ	IQ				
RAVEN					85.15	86.18	81.04	2.951
1	0.35	0.75	1	1	84.82	85.77	80.99	2.913
2	0.30	0.75	1	1	85.07	86.14	80.81	2.957
2	0.35	1.00	1	1	85.24	86.27	81.14	2.936
2	0.35	0.75	2	2	84.92	85.82	**81.33**	2.914
2	**0.35**	**0.75**	**1**	**1**	**85.46**	**86.54**	81.17	**2.962**

Effect of Consistency Policy. All entries in Table 4 share the RAVEN initialization and reward composition, differing only in the policy objective. The Euler-Maruyama (EM) interfaces follow the discretization of Flow-GRPO [41], with σ denoting the auxiliary diffusion noise level and β the reference KL weight. We extend the sweep of σ up to $\sqrt{2}$, the diffusion strength that exactly preserves the bridge variance. The EM interface peaks at $\sigma = 0.8$ and degrades beyond it, as the rollout latents increasingly decode toward noise and drift away from the consistency trajectory used at inference, and the higher noise levels that raise motion do so at a clear cost in quality and semantic alignment. No EM configuration reaches CM-GRPO on total or quality score, and the margin over the strongest EM setting is several times the seed-level standard deviation reported in Appendix B. Human raters likewise prefer CM-GRPO over both RAVEN and the strongest EM setting in Appendix C. This indicates that an auxiliary stochastic process is unnecessary once the policy kernel matches the inference sampler, which also relieves the joint tuning of the diffusion noise and KL weight.

Table 4: Ablation on Consistency Policy.

Policy Interface	Total Score	Qual. Score	Sem. Score	Dyn. Deg.
RAVEN	85.15	86.18	81.04	2.951
+ EM ($\sigma = 0.1$, $\beta = 0$)	85.06	86.10	80.92	2.949
+ EM ($\sigma = 0.4$, $\beta = 0$)	85.15	86.20	80.95	2.949
+ EM ($\sigma = 0.8$, $\beta = 0$)	85.22	86.31	80.84	2.951
+ EM ($\sigma = 0.85$, $\beta = 0$)	84.73	85.83	80.34	2.897
+ EM ($\sigma = 0.9$, $\beta = 0$)	83.73	84.63	80.13	2.928
+ EM ($\sigma = 1.0$, $\beta = 0$)	83.44	84.12	80.73	2.979
+ EM ($\sigma = 1.1$, $\beta = 0$)	83.82	84.81	79.84	**2.988**
+ EM ($\sigma = 1.2$, $\beta = 0$)	84.17	85.19	80.07	2.953
+ EM ($\sigma = \sqrt{2}$, $\beta = 0$)	81.42	82.74	76.11	2.837
+ EM ($\sigma = 0.1$, $\beta = 0.004$)	85.03	86.07	80.86	2.947
+ EM ($\sigma = 0.4$, $\beta = 0.004$)	85.14	86.16	81.09	2.950
+ EM ($\sigma = 0.8$, $\beta = 0.004$)	85.27	86.26	**81.29**	2.950
+ CM-GRPO (Ours)	**85.46**	**86.54**	81.17	2.962

4.3 Scaling to Larger Generators and Longer Horizons

Model-Size Scaling. Since no prior causal distillation baseline is released at 14B, we train both RAVEN and Self Forcing on Wan2.1-T2V-14B [13] under the identical initialization and recipe, following the controlled comparison of Table 2. As shown in Table 5, the stronger base model saturates VBench and narrows the total score margin, yet RAVEN still raises dynamic degree by about 11% and is preferred by human raters in nearly two of every three overall comparisons. This preference is higher than at 1.3B (Appendix C), indicating that the benefit of history supervision grows rather than fades with model capacity.

Table 5: Model-size scaling on Wan2.1-T2V-14B.

Method	Total Score	Dyn. Deg.	Qual. Pref.	Sem. Pref.	Over. Pref.
Self Forcing	84.56	2.474	41.0%	42.5%	35.0%
RAVEN	**84.75**	**2.745**	**59.0%**	**57.5%**	**65.0%**

Long-Horizon Extrapolation. Table 6 extends the 5-second checkpoints to 60 seconds without any long-horizon training. Under sliding-window inference and the full VBench-Long protocol, RAVEN matches Causal Forcing on quality score while leading clearly on total and semantic scores, and human raters favor RAVEN on every dimension. The failure mode of the detached baseline at long horizons is semantic drift away from the prompt, which is exactly the gap that supervising self-generated history is designed to close. We further compare against the long-video systems Rolling Forcing [32] and LongLive [34] on the 128 MovieGen prompts [9] following [56, 57], where RAVEN adopts the same attention sink at inference while both baselines are trained with their long-video mechanisms. RAVEN still leads both systems on average quality, dynamic degree, and user preference.

Training Cost. RAVEN replaces only the Self-Forcing DMD stage shared with Causal Forcing and completes it with about 30 fewer GPU hours, since its generator step reuses the fake-score rollout in a single forward pass, and the full cost breakdown is given in Appendix B.

Table 6: 60-second extrapolation of 5-second checkpoints. The top part uses sliding-window inference under the full VBench-Long protocol, and the bottom part uses attention-sink inference on the 128 MovieGen prompts, where Avg. Score averages the VBench-Long quality dimensions. Preference rates are pairwise human win rates within each block following Appendix C.

Sliding Window	Total Score	Qual. Score	Sem. Score	Qual. Pref.	Sem. Pref.	Over. Pref.
Causal Forcing [36]	77.79	82.30	59.76	15.5%	15.0%	17.0%
RAVEN	**79.94**	82.30	**70.52**	**84.5%**	**85.0%**	**83.0%**

Attention Sink	Avg. Score	Dyn. Deg.	Qual. Pref.	Sem. Pref.	Over. Pref.
Rolling Forcing [32]	76.68	2.936	42.6%	41.0%	41.4%
RAVEN	**81.92**	**2.989**	**57.4%**	**59.0%**	**58.6%**
LongLive [34]	77.94	2.887	44.5%	43.8%	41.0%
RAVEN	**81.92**	**2.989**	**55.5%**	**56.2%**	**59.0%**

5 Conclusion

We presented RAVEN, a training-time test framework that repacks each self rollout into an interleaved sequence of clean historical endpoints and noisy denoising states so that supervision propagates through the cached history used during autoregressive extrapolation. We further proposed CM-GRPO, which formulates the policy objective directly on the consistency transition kernel used at inference rather than on an auxiliary Euler-Maruyama process. Together they surpass recent causal video distillation baselines across quality, semantic, and dynamic dimensions, with their combination yielding the strongest results on every metric considered, and RAVEN retains its advantage at 14B and over 60 seconds. Both formulations also admit broader scope than the setting evaluated here, and we discuss their generalizability and limitations in greater detail in Appendix D.

Acknowledgments and Disclosure of Funding

We thank Fei Ni and Changrui Chen for their valuable discussions and assistance. J. Deng was supported by the NVIDIA Academic Grant. The authors acknowledge the use of resources provided by the Isambard-AI National AI Research Resource (AIRR). Isambard-AI [127] is operated by the University of Bristol and is funded by the UK Government's Department for Science, Innovation and Technology (DSIT) via UK Research and Innovation; and the Science and Technology Facilities Council [ST/AIRR/I-A-I/1023]. The authors declare no competing financial interests.

References

[1] Vladimir Arkhipkin, Vladimir Korviakov, Nikolai Gerasimenko, Denis Parkhomenko, Viacheslav Vasilev, Alexey Letunovskiy, Nikolai Vaulin, Maria Kovaleva, Ivan Kirillov, Lev Novitskiy, Denis Koposov, Nikita Kiselev, et al. Kandinsky 5.0: A family of foundation models for image and video generation. *arXiv preprint arXiv:2511.14993*, 2025.

[2] Junsong Chen, Yuyang Zhao, Jincheng Yu, Ruihang Chu, Junyu Chen, Shuai Yang, Xianbang Wang, Yicheng Pan, Daquan Zhou, Huan Ling, Haozhe Liu, Hongwei Yi, et al. Sana-video: Efficient video generation with block linear diffusion transformer. In *ICLR*, 2026.

[3] Yu Gao, Haoyuan Guo, Tuyen Hoang, Weilin Huang, Lu Jiang, Fangyuan Kong, Huixia Li, Jiashi Li, Liang Li, Xiaojie Li, Xunsong Li, Yifu Li, et al. Seedance 1.0: Exploring the boundaries of video generation models. *arXiv preprint arXiv:2506.09113*, 2025.

[4] Genmo. Mochi 1: Ai video generator. `https://www.genmo.ai/blog/mochi-1-a-new-sota-in-open-text-to-video`, 2024.

[5] Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, Poriya Panet, Sapir Weissbuch, et al. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

[6] Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, Eitan Richardson, Guy Shiran, et al. Ltx-2: Efficient joint audio-visual foundation model. *arXiv preprint arXiv:2601.03233*, 2026.

[7] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, Kathrina Wu, Qin Lin, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2025.

[8] Guoqing Ma, Haoyang Huang, Kun Yan, Liangyu Chen, Nan Duan, Shengming Yin, Changyi Wan, Ranchen Ming, Xiaoniu Song, Xing Chen, Yu Zhou, Deshan Sun, et al. Step-video-t2v technical report: The practice, challenges, and future of video foundation model. *arXiv preprint arXiv:2502.10248*, 2025.

[9] Meta. Movie gen: A cast of media foundation models. `https://ai.meta.com/static-resource/movie-gen-research-paper`, 2024.

[10] NVIDIA. Cosmos world foundation model platform for physical ai. `https://research.nvidia.com/publication/2025-01_cosmos-world-foundation-model-platform-physical-ai`, 2025.

[11] Team Seedance, Heyi Chen, Siyan Chen, Xin Chen, Yanfei Chen, Ying Chen, Zhuo Chen, Feng Cheng, Tianheng Cheng, Xinqi Cheng, Xuyan Chi, Jian Cong, et al. Seedance 1.5 pro: A native audio-visual joint generation foundation model. *arXiv preprint arXiv:2512.13507*, 2025.

[12] Team Seedance, De Chen, Liyang Chen, Xin Chen, Ying Chen, Zhuo Chen, Zhuowei Chen, Feng Cheng, Tianheng Cheng, Yufeng Cheng, Mojie Chi, Xuyan Chi, et al. Seedance 2.0: Advancing video generation for world complexity. *arXiv preprint arXiv:2604.14148*, 2026.

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

[14] Bing Wu, Chang Zou, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Jack Peng, Jianbing Wu, Jiangfeng Xiong, Jie Jiang, Linus, Patrol, et al. Hunyuanvideo 1.5 technical report. *arXiv preprint arXiv:2511.18870*, 2025.

[15] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Xiaotao Gu, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *ICLR*, 2025.

[16] Sand ai, Hansi Teng, Hongyu Jia, Lei Sun, Lingzhi Li, Maolin Li, Mingqiu Tang, Shuai Han, Tianning Zhang, W. Q. Zhang, Weifeng Luo, Xiaoyang Kang, et al. Magi-1: Autoregressive video generation at scale. *arXiv preprint arXiv:2505.13211*, 2025.

[17] Xingjian Bai, Guande He, Zhengqi Li, Eli Shechtman, Xun Huang, and Zongze Wu. Causality in video diffusers is separable from denoising. *arXiv preprint arXiv:2602.10095*, 2026.

[18] Guibin Chen, Dixuan Lin, Jiangping Yang, Chunze Lin, Junchen Zhu, Mingyuan Fan, Hao Zhang, Sheng Chen, Zheng Chen, Chengcheng Ma, Weiming Xiong, Wei Wang, et al. Skyreels-v2: Infinite-length film generative model. *arXiv preprint arXiv:2504.13074*, 2025.

[19] Haoge Deng, Ting Pan, Haiwen Diao, Zhengxiong Luo, Yufeng Cui, Huchuan Lu, Shiguang Shan, Yonggang Qi, and Xinlong Wang. Autoregressive video generation without vector quantization. In *ICLR*, 2025.

[20] Yuwei Guo, Ceyuan Yang, Hao He, Yang Zhao, Meng Wei, Zhenheng Yang, Weilin Huang, and Dahua Lin. End-to-end training for autoregressive video diffusion via self-resampling. *arXiv preprint arXiv:2512.15702*, 2025.

[21] Junchao Huang, Ziyang Ye, Xinting Hu, Tianyu He, Guiyu Zhang, Shaoshuai Shi, Jiang Bian, and Li Jiang. Live: Long-horizon interactive video world modeling. *arXiv preprint arXiv:2602.03747*, 2026.

[22] Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. In *ICLR*, 2025.

[23] Wuyang Li, Wentao Pan, Po-Chien Luan, Yang Gao, and Alexandre Alahi. Stable video infinity: Infinite-length video generation with error recycling. In *ICLR*, 2026.

[24] Jinlai Liu, Jian Han, Bin Yan, Hui Wu, Fengda Zhu, Xing Wang, Yi Jiang, Bingyue Peng, and Zehuan Yuan. Infinitystar: Unified spacetime autoregressive modeling for visual generation. In *NeurIPS*, 2025.

[25] Ryan Po, Eric Ryan Chan, Changan Chen, and Gordon Wetzstein. Bagger: Backwards aggregation for mitigating drift in autoregressive video diffusion models. *arXiv preprint arXiv:2512.12080*, 2025.

[26] Xiaofei Wu, Guozhen Zhang, Zhiyong Xu, Yuan Zhou, Qinglin Lu, and Xuming He. Pack and force your memory: Long-form and consistent video generation. *arXiv preprint arXiv:2510.01784*, 2025.

[27] Xunzhi Xiang, Yabo Chen, Guiyu Zhang, Zhongyu Wang, Zhe Gao, Quanming Xiang, Gonghu Shang, Junqi Liu, Haibin Huang, Yang Gao, Chi Zhang, Qi Fan, et al. Macro-from-micro planning for high-quality and parallelized autoregressive long video generation. *arXiv preprint arXiv:2508.03334*, 2025.

[28] Shenghai Yuan, Yuanyang Yin, Zongjian Li, Xinwei Huang, Xiao Yang, and Li Yuan. Helios: Real real-time long video generation model. *arXiv preprint arXiv:2603.04379*, 2026.

[29] Zeyu Zhang, Shuning Chang, Yuanyu He, Yizeng Han, Jiasheng Tang, Fan Wang, and Bohan Zhuang. Blockvid: Block diffusion for high-quality and consistent minute-long video generation. *arXiv preprint arXiv:2511.22973*, 2025.

[30] Lvmin Zhang, Shengqu Cai, Muyang Li, Chong Zeng, Beijia Lu, Anyi Rao, Song Han, Gordon Wetzstein, and Maneesh Agrawala. Pretraining frame preservation for lightweight autoregressive video history embedding. *arXiv preprint arXiv:2512.23851*, 2026.

[31] Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *NeurIPS*, 2025.

[32] Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling forcing: Autoregressive long video diffusion in real time. In *ICLR*, 2026.

[33] Yunhong Lu, Yanhong Zeng, Haobo Li, Hao Ouyang, Qiuyu Wang, Ka Leong Cheng, Jiapeng Zhu, Hengyuan Cao, Zhipeng Zhang, Xing Zhu, Yujun Shen, and Min Zhang. Reward forcing: Efficient streaming video generation with rewarded distribution matching distillation. In *CVPR*, 2026.

[34] Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, Song Han, and Yukang Chen. Longlive: Real-time interactive long video generation. In *ICLR*, 2026.

[35] Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *CVPR*, 2025.

[36] Hongzhou Zhu, Min Zhao, Guande He, Hang Su, Chongxuan Li, and Jun Zhu. Causal forcing: Autoregressive diffusion distillation done right for high-quality real-time interactive video generation. In *ICML*, 2026.

[37] Boyuan Chen, Diego Marti Monso, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *NeurIPS*, 2024.

[38] Kiwhan Song, Boyuan Chen, Max Simchowitz, Yilun Du, Russ Tedrake, and Vincent Sitzmann. History-guided video diffusion. In *ICML*, 2025.

[39] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *CVPR*, pages 6613–6623, 2024.

[40] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T. Freeman. Improved distribution matching distillation for fast image synthesis. In *NeurIPS*, 2024.

[41] 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. In *NeurIPS*, 2025.

[42] Shuo Chen, Cong Wei, Sun Sun, Ping Nie, Kai Zhou, Ge Zhang, Ming-Hsuan Yang, and Wenhu Chen. Context forcing: Consistent autoregressive video generation with long context. *arXiv preprint arXiv:2602.06028*, 2026.

[43] Justin Cui, Jie Wu, Ming Li, Tao Yang, Xiaojie Li, Rui Wang, Andrew Bai, Yuanhao Ban, and Cho-Jui Hsieh. Self-forcing++: Towards minute-scale high-quality video generation. *arXiv preprint arXiv:2510.02283*, 2025.

[44] Jinxiu Liu, Xuanming Liu, Kangfu Mei, Yandong Wen, Ming-Hsuan Yang, and Weiyang Liu. Streaming autoregressive video generation via diagonal distillation. In *ICLR*, 2026.

[45] Kai Zou, Dian Zheng, Hongbo Liu, Tiankai Hang, Bin Liu, and Nenghai Yu. Hiar: Efficient autoregressive long video generation via hierarchical denoising. *arXiv preprint arXiv:2603.08703*, 2026.

[46] Hanmo Chen, Chenghao Xu, Xu Yang, Xuan Chen, and Cheng Deng. Past- and future-informed kv cache policy with salience estimation in autoregressive video diffusion. *arXiv preprint arXiv:2601.21896*, 2026.

[47] Sihui Ji, Xi Chen, Shuai Yang, Xin Tao, Pengfei Wan, and Hengshuang Zhao. Memflow: Flowing adaptive memory for consistent and efficient long video narratives. *arXiv preprint arXiv:2512.14699*, 2025.

[48] Yicheng Ji, Zhizhou Zhong, Jun Zhang, Qin Yang, XiTai Jin, Ying Qin, Wenhan Luo, Shuiyang Mao, Wei Liu, and Huan Li. Forcing-kv: Hybrid kv cache compression for efficient autoregressive video diffusion models. *arXiv preprint arXiv:2605.09681*, 2026.

[49] Joonghyuk Shin, Zhengqi Li, Richard Zhang, Jun-Yan Zhu, Jaesik Park, Eli Shechtman, and Xun Huang. Motionstream: Real-time video generation with interactive motion controls. In *ICLR*, 2026.

[50] Yifei Yu, Xiaoshan Wu, Xinting Hu, Tao Hu, Yangtian Sun, Xiaoyang Lyu, Bo Wang, Lin Ma, Yuewen Ma, Zhongrui Wang, and Xiaojuan Qi. Videossm: Autoregressive long video generation with hybrid state-space memory. *arXiv preprint arXiv:2512.04519*, 2025.

[51] Tianrui Zhu, Shiyi Zhang, Zhirui Sun, Jingqi Tian, and Yansong Tang. Memorize-and-generate: Towards long-term consistency in real-time video generation. *arXiv preprint arXiv:2512.18741*, 2025.

[52] Justin Cui, Jie Wu, Ming Li, Tao Yang, Xiaojie Li, Rui Wang, Andrew Bai, Yuanhao Ban, and Cho-Jui Hsieh. Lol: Longer than longer, scaling video generation to hour. *arXiv preprint arXiv:2601.16914*, 2026.

[53] Jia Li, Xiaomeng Fu, Xurui Peng, Weifeng Chen, Youwei Zheng, Tianyu Zhao, Jiexi Wang, Fangmin Chen, Xing Wang, and Hayden Kwok-Hay So. Train short, inference long: Training-free horizon extension for autoregressive video generation. *arXiv preprint arXiv:2602.14027*, 2026.

[54] Xunzhi Xiang, Zixuan Duan, Guiyu Zhang, Haiyu Zhang, Zhe Gao, Junta Wu, Shaofeng Zhang, Tengfei Wang, Qi Fan, and Chunchao Guo. Pathwise test-time correction for autoregressive long video generation. *arXiv preprint arXiv:2602.05871*, 2026.

[55] Hidir Yesiltepe, Tuna Han Salih Meral, Adil Kaan Akan, Kaan Oktay, and Pinar Yanardag. Infinity-rope: Action-controllable infinite video generation emerges from autoregressive self-rollout. In *CVPR*, 2026.

[56] Jung Yi, Wooseok Jang, Paul Hyunbin Cho, Jisu Nam, Heeji Yoon, and Seungryong Kim. Deep forcing: Training-free long video generation with deep sink and participative compression. In *ICML*, 2026.

[57] Zengqun Zhao, Yanzuo Lu, Ziquan Liu, Jifei Song, Jiankang Deng, and Ioannis Patras. Relax forcing: Relaxed kv-memory for consistent long video generation. *arXiv preprint arXiv:2603.21366*, 2026.

[58] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *ICLR*, 2024.

[59] Yibin Wang, Zhimin Li, Yuhang Zang, Yujie Zhou, Jiazi Bu, Chunyu Wang, Qinglin Lu, Cheng Jin, and Jiaqi Wang. Pref-grpo: Pairwise preference reward-based grpo for stable text-to-image reinforcement learning. *arXiv preprint arXiv:2508.20751*, 2026.

[60] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. In *NeurIPS*, 2023.

[61] Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. Diffusionnft: Online diffusion reinforcement with forward process. In *ICLR*, 2026.

[62] Xiaoxiao Ma, Haibo Qiu, Guohui Zhang, Zhixiong Zeng, Siqi Yang, Lin Ma, and Feng Zhao. Stage: Stable and generalizable grpo for autoregressive image generation. *arXiv preprint arXiv:2509.25027*, 2025.

[63] Zehan Wang, Tengfei Wang, Haiyu Zhang, Xuhui Zuo, Junta Wu, Haoyuan Wang, Wenqiang Sun, Zhenwei Wang, Chenjie Cao, Hengshuang Zhao, Chunchao Guo, and Zhou Zhao. Worldcompass: Reinforcement learning for long-horizon world models. *arXiv preprint arXiv:2602.09022*, 2026.

[64] Jialong Wu, Shaofeng Yin, Ningya Feng, and Mingsheng Long. Rlvr-world: Training world models with reinforcement learning. In *ICLR*, 2025.

[65] Yang Ye, Tianyu He, Shuo Yang, and Jiang Bian. Reinforcement learning with inverse rewards for world model post-training. *arXiv preprint arXiv:2509.23958*, 2025.

[66] Songchun Zhang, Zeyue Xue, Siming Fu, Jie Huang, Xianghao Kong, Y. Ma, Haoyang Huang, Nan Duan, and Anyi Rao. Astrolabe: Steering forward-process reinforcement learning for distilled autoregressive video models. *arXiv preprint arXiv:2603.17051*, 2026.

[67] Kesen Zhao, Jiaxin Shi, Beier Zhu, Junbao Zhou, Xiaolong Shen, Yuan Zhou, Qianru Sun, and Hanwang Zhang. Real-time motion-controllable autoregressive video diffusion. In *ICLR*, 2026.

[68] Guanjie Chen, Shirui Huang, Kai Liu, Jianchen Zhu, Xiaoye Qu, Peng Chen, Yu Cheng, and Yifu Sun. Flash-dmd: Towards high-fidelity few-step image generation with efficient distillation and joint reinforcement learning. *arXiv preprint arXiv:2511.20549*, 2025.

[69] Xiefan Guo, Xinzhu Ma, Haoxiang Ma, Zihao Zhou, and Di Huang. Erudiff: Refactoring knowledge in diffusion models for advanced text-to-image synthesis. *arXiv preprint arXiv:2603.20828*, 2026.

[70] Yihong Luo, Tianyang Hu, Weijian Luo, and Jing Tang. Tdm-r1: Reinforcing few-step diffusion models with non-differentiable reward. *arXiv preprint arXiv:2603.07700*, 2026.

[71] Owen Oertell, Jonathan D. Chang, Yiyi Zhang, Kianté Brantley, and Wen Sun. Rl for consistency models: Faster reward guided text-to-image generation. In *ICML*, 2024.

[72] Haoran He, Yuxiao Ye, Jie Liu, Jiajun Liang, Zhiyong Wang, Ziyang Yuan, Xintao Wang, Hangyu Mao, Pengfei Wan, and Ling Pan. Gardo: Reinforcing diffusion models without reward hacking. *arXiv preprint arXiv:2512.24138*, 2025.

[73] Jie Liu, Zilyu Ye, Linxiao Yuan, Shenhan Zhu, Yu Gao, Jie Wu, Kunchang Li, Xionghui Wang, Xiaonan Nie, Weilin Huang, and Wanli Ouyang. Unigrpo: Unified policy optimization for reasoning-driven visual generation. *arXiv preprint arXiv:2603.23500*, 2026.

[74] Haotian Ye, Kaiwen Zheng, Jiashu Xu, Puheng Li, Huayu Chen, Jiaqi Han, Sheng Liu, Qinsheng Zhang, Hanzi Mao, Zekun Hao, Prithvijit Chattopadhyay, Dinghao Yang, et al. Data-regularized reinforcement learning for diffusion models at scale. *arXiv preprint arXiv:2512.04332*, 2025.

[75] Yuanzhi Zhu, Xi Wang, Stéphane Lathuilière, and Vicky Kalogeiton. Diffusion reinforcement learning via centered reward distillation. *arXiv preprint arXiv:2603.14128*, 2026.

[76] Dailan He, Guanlin Feng, Xingtong Ge, Yazhe Niu, Yi Zhang, Bingqi Ma, Guanglu Song, Yu Liu, and Hongsheng Li. Neighbor grpo: Contrastive ode policy optimization aligns flow models. In *CVPR*, 2026.

[77] Yihong Luo, Tianyang Hu, and Jing Tang. Reinforcing diffusion models by direct group preference optimization. In *ICLR*, 2026.

[78] Jiayuan Sheng, Hanyang Zhao, Haoxian Chen, David D. Yao, and Wenpin Tang. Understanding sampler stochasticity in training diffusion models for rlhf. *arXiv preprint arXiv:2510.10767*, 2025.

[79] Feng Wang and Zihao Yu. Coefficients-preserving sampling for reinforcement learning with flow matching. *arXiv preprint arXiv:2509.05952*, 2025.

[80] Shaomeng Wang, He Wang, Longquan Dai, and Jinhui Tang. Pc-flow: Preference alignment in flow matching via classifier. In *AAAI*, 2026.

[81] Shengjun Zhang, Zhang Zhang, Chensheng Dai, and Yueqi Duan. E-grpo: High entropy steps drive effective reinforcement learning for flow models. In *CVPR Findings*, 2026.

[82] Ruicheng Zhang, Kaixi Cong, Jun Zhou, Zhizhou Zhong, Zunnan Xu, Shuiyang Mao, Wei Liu, and Xiu Li. Kvpo: Ode-native grpo for autoregressive video alignment via kv semantic exploration. *arXiv preprint arXiv:2605.14278*, 2026.

[83] Mingzhe Zheng, Weijie Kong, Yue Wu, Dengyang Jiang, Yue Ma, Xuanhua He, Bin Lin, Kaixiong Gong, Zhao Zhong, Liefeng Bo, Qifeng Chen, and Harry Yang. Manifold-aware exploration for reinforcement learning in video generation. *arXiv preprint arXiv:2603.21872*, 2026.

[84] Kaijie Chen, Zhiyang Xu, Ying Shen, Zihao Lin, Yuguang Yao, and Lifu Huang. Superflow: Training flow matching models with rl on the fly. *arXiv preprint arXiv:2512.17951*, 2026.

[85] Zheng Ding and Weirui Ye. Treegrpo: Tree-advantage grpo for online rl post-training of diffusion models. In *ICLR*, 2026.

[86] Xiaolong Fu, Lichen Ma, Zipeng Guo, Gaojing Zhou, Chongxiao Wang, ShiPing Dong, Shizhe Zhou, Shizhe Zhou, Ximan Liu, Jingling Fu, Tan Lit Sin, Yu Shi, et al. Dynamic-treerpo: Breaking the independent trajectory bottleneck with structured sampling. *arXiv preprint arXiv:2509.23352*, 2025.

[87] Xiaoxuan He, Siming Fu, Yuke Zhao, Wanli Li, Jian Yang, Dacheng Yin, Fengyun Rao, and Bo Zhang. Tempflow-grpo: When timing matters for grpo in flow models. In *ICLR*, 2026.

[88] Yuming Li, Yikai Wang, Yuying Zhu, Zhongyu Zhao, Ming Lu, Qi She, and Shanghang Zhang. Branch-grpo: Stable and efficient grpo with structured branching in diffusion models. In *ICLR*, 2026.

[89] Zhanhao Liang, Tao Yang, Jie Wu, Chengjian Feng, and Liang Zheng. Leapalign: Post-training flow matching models at any generation step by building two-step trajectories. *arXiv preprint arXiv:2604.15311*, 2026.

[90] Yifu Luo, Penghui Du, Bo Li, Sinan Du, Tiantian Zhang, Yongzhe Chang, Kai Wu, Kun Gai, and Xueqian Wang. Sample by step, optimize by chunk: Chunk-level grpo for text-to-image generation. *arXiv preprint arXiv:2510.21583*, 2025.

[91] Qiang Lyu, Zicong Chen, Chongxiao Wang, Haolin Shi, Shibo Gao, Ran Piao, Youwei Zeng, Jianlou Si, Fei Ding, Jing Li, Chun Pong Lau, and Weiqiang Wang. Multi-grpo: Multi-group advantage estimation for text-to-image generation with tree-based trajectories and multiple rewards. *arXiv preprint arXiv:2512.00743*, 2025.

[92] David McAllister, Miika Aittala, Tero Karras, Janne Hellsten, Angjoo Kanazawa, Timo Aila, and Samuli Laine. Finite difference flow optimization for rl post-training of text-to-image models. *arXiv preprint arXiv:2603.12893*, 2026.

[93] Yash Savani, Branislav Kveton, Yuchen Liu, Yilin Wang, Jing Shi, Subhojyoti Mukherjee, Nikos Vlassis, and Krishna Kumar Singh. Stepwise credit assignment for grpo on flow-matching models. In *CVPR*, 2026.

[94] Yawen Shao, Jie Xiao, Kai Zhu, Yu Liu, Wei Zhai, Yang Cao, and Zheng-Jun Zha. Anchoring values in temporal and group dimensions for flow matching model alignment. *arXiv preprint arXiv:2512.12387*, 2025.

[95] Sophia Tang, Yuchen Zhu, Molei Tao, and Pranam Chatterjee. Tr2-d2: Tree search guided trajectory-aware fine-tuning for discrete diffusion. *arXiv preprint arXiv:2509.25171*, 2025.

[96] Yunze Tong, Mushui Liu, Canyu Zhao, Wanggui He, Shiyi Zhang, Hongwei Zhang, Peng Zhang, Jinlong Liu, Ju Huang, Jiamang Wang, Hao Jiang, and Pipei Huang. Alleviating sparse rewards by modeling step-wise and long-term sampling effects in flow-based grpo. *arXiv preprint arXiv:2602.06422*, 2026.

[97] Jing Wang, Jiajun Liang, Jie Liu, Henglin Liu, Gongye Liu, Jun Zheng, Wanyuan Pang, Ao Ma, Zhenyu Xie, Xintao Wang, Meng Wang, Pengfei Wan, et al. Grpo-guard: Mitigating implicit over-optimization in flow matching via regulated clipping. *arXiv preprint arXiv:2510.22319*, 2025.

[98] Zhixiong Yue, Zixuan Ni, Feiyang Ye, Jinshan Zhang, Sheng Shen, and Zhenpeng Mi. Know your step: Faster and better alignment for flow matching models via step-aware advantages. *arXiv preprint arXiv:2602.01591*, 2026.

[99] Liyu Zhang, Kehan Li, Tingrui Han, Tao Zhao, Yuxuan Sheng, Shibo He, and Chao Li. Op-grpo: Efficient off-policy grpo for flow-matching models. *arXiv preprint arXiv:2604.04142*, 2026.

[100] Yujie Zhou, Pengyang Ling, Jiazi Bu, Yibin Wang, Yuhang Zang, Jiaqi Wang, Li Niu, and Guangtao Zhai. Fine-grained grpo for precise preference alignment in flow models. In *CVPR*, 2026.

[101] Shanchuan Lin, Anran Wang, and Xiao Yang. Sdxl-lightning: Progressive adversarial diffusion distillation. *arXiv preprint arXiv:2402.13929*, 2024.

[102] Yanzuo Lu, Xin Xia, Manlin Zhang, Huafeng Kuang, Jianbin Zheng, Yuxi Ren, and Xuefeng Xiao. Hyper-bagel: A unified acceleration framework for multimodal understanding and generation. *arXiv preprint arXiv:2509.18824*, 2025.

[103] Yanzuo Lu, Yuxi Ren, Xin Xia, Shanchuan Lin, Xing Wang, Xuefeng Xiao, Andy J Ma, Xiaohua Xie, and Jian-Huang Lai. Adversarial distribution matching for diffusion distillation towards efficient image and video synthesis. In *ICCV*, 2025.

[104] Yuxi Ren, Xin Xia, Yanzuo Lu, Jiacheng Zhang, Jie Wu, Pan Xie, Xing Wang, and Xuefeng Xiao. Hyper-sd: Trajectory segmented consistency model for efficient image synthesis. In *NeurIPS*, 2024.

[105] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *ICLR*, 2022.

[106] Fu-Yun Wang, Zhaoyang Huang, Alexander William Bergman, Dazhong Shen, Peng Gao, Michael Lingelbach, Keqiang Sun, Weikang Bian, Guanglu Song, Yu Liu, Hongsheng Li, and Xiaogang Wang. Phased consistency model. In *NeurIPS*, 2024.

[107] Hanshu Yan, Xingchao Liu, Jiachun Pan, Jun Hao Liew, Qiang Liu, and Jiashi Feng. Perflow: Piecewise rectified flow as universal plug-and-play accelerator. In *NeurIPS*, 2024.

[108] Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. In *ICLR*, 2024.

[109] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

[110] Simian Luo, Yiqin Tan, Suraj Patil, Daniel Gu, Patrick von Platen, Apolinário Passos, Longbo Huang, Jian Li, and Hang Zhao. Lcm-lora: A universal stable-diffusion acceleration module. *arXiv preprint arXiv:2311.05556*, 2023.

[111] Weijian Luo, Zemin Huang, Zhengyang Geng, J. Zico Kolter, and Guo-jun Qi. One-step diffusion distillation through score implicit matching. In *NeurIPS*, 2024.

[112] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *ICML*, 2023.

[113] Jianbin Zheng, Minghui Hu, Zhongyi Fan, Chaoyue Wang, Changxing Ding, Dacheng Tao, and Tat-Jen Cham. Trajectory consistency distillation: Improved latent consistency distillation by semi-linear consistency function with trajectory mapping. *arXiv preprint arXiv:2402.19159*, 2024.

[114] Mingyuan Zhou, Huangjie Zheng, Yi Gu, Zhendong Wang, and Hai Huang. Adversarial score identity distillation: Rapidly surpassing the teacher in one step. In *ICLR*, 2025.

[115] Mingyuan Zhou, Huangjie Zheng, Zhendong Wang, Mingzhang Yin, and Hai Huang. Score identity distillation: Exponentially fast distillation of pretrained diffusion models for one-step generation. In *ICML*, 2024.

[116] Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. Eagle-3: Scaling up inference acceleration of large language models via training-time test. In *NeurIPS*, 2025.

[117] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[118] Jie Liu, Gongye Liu, Jiajun Liang, Ziyang Yuan, Xiaokun Liu, Mingwu Zheng, Xiele Wu, Qiulin Wang, Menghan Xia, Xintao Wang, Xiaohong Liu, Fei Yang, et al. Improving video generation with human feedback. In *NeurIPS*, 2025.

[119] Yibin Wang, Yuhang Zang, Hao Li, Cheng Jin, and Jiaqi Wang. Unified reward model for multimodal understanding and generation. *arXiv preprint arXiv:2503.05236*, 2026.

[120] Gongye Liu, Bo Yang, Yida Zhi, Zhizhou Zhong, Lei Ke, Didan Deng, Han Gao, Yongxiang Huang, Kaihao Zhang, Hongbo Fu, and Wenhan Luo. Beyond vlm-based rewards: Diffusion-native latent reward modeling. In *ICML*, 2026.

[121] Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *ICCV*, pages 5128–5137, 2021.

[122] LAION-AI. aesthetic-predictor. `https://github.com/LAION-AI/aesthetic-predictor`, 2022.

[123] Zhen Li, Zuo-Liang Zhu, Ling-Hao Han, Qibin Hou, Chun-Le Guo, and Ming-Ming Cheng. Amt: All-pairs multi-field transforms for efficient frame interpolation. In *CVPR*, pages 9801–9810, 2023.

[124] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, volume 12347, pages 402–419, 2020.

[125] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, et al. Vbench: Comprehensive benchmark suite for video generative models. In *CVPR*, 2024.

[126] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024.

[127] Simon McIntosh-Smith, Sadaf Alam, and Christopher Woods. Isambard-ai: a leadership-class supercomputer optimised specifically for artificial intelligence. *arXiv preprint arXiv:2410.11199*, 2024.

[128] Wenhao Wang and Yi Yang. Vidprom: A million-scale real prompt-gallery dataset for text-to-video diffusion models. In *NeurIPS Datasets and Benchmarks Track*, 2024.

[129] Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. In *ICLR*, 2025.

[130] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Real-time intermediate flow estimation for video frame interpolation. In *ECCV*, 2022.

[131] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022.

[132] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[133] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[134] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[135] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, 2023.

[136] Ronglai Zuo, Rolandos Alexandros Potamias, Qi Sun, Evangelos Ververas, Jiankang Deng, and Stefanos Zafeiriou. Madis: Taming masked diffusion language models for sign language generation. *arXiv preprint arXiv:2601.19577*, 2026.

[137] Fei Ni, Zhuo Chen, Yifu Yuan, Zibin Dong, Xianze Yao, Shan Luo, Jianye Hao, Jiankang Deng, and Stefanos Zafeiriou. Semanticvla: Towards semantic reasoning over action memorization via synergistic explicit trace and latent action planning. In *CVPR*, pages 12237–12247, 2026.

Algorithm 1 RAVEN training iteration

Require: causal student f_θ, fake-score critic f_ϕ, bidirectional teacher f_ψ
Require: text condition c, chunk count T, noise schedule (α_n, σ_n)
Require: sampling timesteps $\tau_1 > \cdots > \tau_K = 0$
Require: chunk-wise weighting function g_η, learning rates η_θ and η_ϕ
Require: iteration index i, TTUR ratio r (critic-to-generator update frequency)

Stage 1: Self Rollout *no gradient through f_θ*

for $t = 1, \ldots, T$ **do**
  sample initial noisy state $\hat{z}_t^{(\tau_1)} \sim \mathcal{N}(0, I)$
  for $k = 1, \ldots, K-1$ **do**
    predict clean endpoint $\hat{x}_t^{(k)} \leftarrow \mathrm{Endpoint}\big(f_\theta(\hat{z}_t^{(\tau_k)}, \tau_k, c, h_t)\big)$
    sample $\epsilon \sim \mathcal{N}(0, I)$
    consistency transition $\hat{z}_t^{(\tau_{k+1})} \leftarrow \alpha_{\tau_{k+1}} \hat{x}_t^{(k)} + \sigma_{\tau_{k+1}} \epsilon$ ▷ Eq. (6)
  end for
  final endpoint $\hat{x}_t \leftarrow \mathrm{Endpoint}\big(f_\theta(\hat{z}_t^{(\tau_K)}, \tau_K, c, h_t)\big)$
  update KV cache with $\hat{x}_t$ so that $h_{t+1} = \mathcal{H}(\hat{x}_{\le t})$
end for

Stage 2: Fake-Score Step

concatenate the rollout endpoints into the full clean sequence $\hat{x} = (\hat{x}_1, \ldots, \hat{x}_T)$
sample $n \sim \mathcal{U}(0, 1)$ and $\epsilon \sim \mathcal{N}(0, I)$
perturb $z^{(n)} \leftarrow \alpha_n \hat{x} + \sigma_n \epsilon$
bidirectional forward pass of f_ϕ on $z^{(n)}$
update critic $\phi \leftarrow \phi - \eta_\phi \nabla_\phi \big\| f_\phi(z^{(n)}, n, c) - \mathrm{Target}(\hat{x}) \big\|^2$

Stage 3: Generator Step on $\mathcal{I}_u$ *runs only when $i \bmod r = 0$*

if $i \bmod r = 0$ **then**
  sample sampling level $u \in \{\tau_1, \ldots, \tau_{K-1}\}$
  sample score level $s \sim \mathcal{U}(0, u)$
  assemble interleaved sequence $\mathcal{I}_u$ from the rollout ▷ Eq. (5)
  causal forward of f_θ on $\mathcal{I}_u$, yielding $\hat{x}_\theta$
  sample $\epsilon \sim \mathcal{N}(0, I)$
  consistency transition $\hat{z}^{(s)} \leftarrow \mathrm{sg}\big(\alpha_s \hat{x}_\theta + \sigma_s \epsilon\big)$ ▷ Eq. (6)
  teacher endpoint $\hat{x}_\psi \leftarrow \mathrm{Endpoint}(f_\psi(\hat{z}^{(s)}, s, c))$
  critic endpoint $\hat{x}_\phi \leftarrow \mathrm{Endpoint}(f_\phi(\hat{z}^{(s)}, s, c))$
  per-element DMD loss $\ell \leftarrow \left\| \hat{x}_\theta - \mathrm{sg}\left(\hat{x}_\theta + \dfrac{\hat{x}_\psi - \hat{x}_\phi}{\|\hat{x}_\theta - \hat{x}_\psi\|_1} \right) \right\|^2$
  split ℓ along the chunk axis into per-chunk losses ℓ_t
  compute future participation scores p_t and chunk weights $w_t \propto g_\eta(p_t)$
  chunk-weighted DMD loss $\mathcal{L}_{\mathrm{DMD}} \leftarrow \dfrac{\sum_t w_t \ell_t}{\sum_t w_t m_t}$
  $\theta \leftarrow \theta - \eta_\theta \nabla_\theta \mathcal{L}_{\mathrm{DMD}}$
end if

A Algorithm Formulations

Algorithm 1 formalizes one RAVEN training iteration. The iteration begins with a self rollout of f_θ under the consistency sampler, producing per-chunk denoising trajectories $\{\hat{z}_t^{(\tau_k)}\}$ together with clean endpoints $\hat{x}_t = \hat{z}_t^{(0)}$. The fake-score critic f_ϕ is updated on noised endpoints every iteration, whereas the generator f_θ is updated only once every r iterations on the interleaved sequence $\mathcal{I}_u$, with a reverse-KL score gradient defined by the teacher f_ψ and the updated critic f_ϕ. A single causal forward pass over $\mathcal{I}_u$ then routes gradients from later chunks through the cached history $h_t^{\mathrm{RAVEN}} = \mathcal{H}(\hat{x}_{<t})$ used during extrapolation.

Building on the RAVEN checkpoint, Algorithm 2 formalizes one CM-GRPO training iteration. The iteration draws a group of G independent consistency rollouts under a shared text condition c,

Algorithm 2 CM-GRPO training iteration

Require: consistency generator f_θ initialized from a RAVEN checkpoint, reward composition $\{R_m\}$ with weights $\{\lambda_m\}$
Require: text condition c, group size G, chunk count T, noise schedule (α_n, σ_n)
Require: sampling timesteps $\tau_1 > \cdots > \tau_K = 0$
Require: chunk-wise weighting function g_η, advantage clip $A_{\max}$, learning rate η_θ
Require: normalization stabilization constant ε

Stage 1: Group Rollouts *no gradient through f_θ*

for $i = 1, \dots, G$ **do**
  for $t = 1, \dots, T$ **do**
    sample initial noisy state $\tilde{z}_{i,t}^{(\tau_1)} \sim \mathcal{N}(0, I)$
    for $k = 1, \dots, K-1$ **do**
      predict clean endpoint $\hat{x}_{i,t}^{(k)} \leftarrow \mathrm{Endpoint}\big(f_\theta(\tilde{z}_{i,t}^{(\tau_k)}, \tau_k, c, h_{i,t})\big)$
      sample $\epsilon \sim \mathcal{N}(0, I)$
      consistency transition $\tilde{z}_{i,t}^{(\tau_{k+1})} \leftarrow \alpha_{\tau_{k+1}} \hat{x}_{i,t}^{(k)} + \sigma_{\tau_{k+1}} \epsilon$ ▷ Eq. (6)
    end for
    final endpoint $\hat{x}_{i,t} \leftarrow \mathrm{Endpoint}\big(f_\theta(\tilde{z}_{i,t}^{(\tau_K)}, \tau_K, c, h_{i,t})\big)$
    update KV cache so that $h_{i,t+1} = \mathcal{H}(\hat{x}_{i,\le t})$
  end for
  trajectory endpoint $\hat{x}^i \leftarrow (\hat{x}_{i,1}, \dots, \hat{x}_{i,T})$
end for

Stage 2: Reward Evaluation and Group-Relative Advantages

for $i = 1, \dots, G$ **do**
  evaluate each reward dimension $R_m^i \leftarrow R_m(\hat{x}^i, c)$
end for
per-dimension group normalization $\bar{R}_m^i \leftarrow \big(R_m^i - \mathrm{mean}(\{R_m^j\}_{j=1}^G)\big) \,/\, \big(\mathrm{std}(\{R_m^j\}_{j=1}^G) + \varepsilon\big)$
composite reward $R^i \leftarrow \sum_m \lambda_m \bar{R}_m^i \,/\, \sum_m |\lambda_m|$
group-relative advantage $\hat{A}_i \leftarrow \big(R^i - \mathrm{mean}(\{R^j\}_{j=1}^G)\big) \,/\, \big(\mathrm{std}(\{R^j\}_{j=1}^G) + \varepsilon\big)$
clip advantage $\hat{A}_i \leftarrow \mathrm{clip}(\hat{A}_i, -A_{\max}, A_{\max})$

Stage 3: Policy Update on Sampled Transitions

for $i = 1, \dots, G$ **do**
  sample $k_i \sim \mathcal{U}\{1, \dots, K-1\}$ and let $u_i \leftarrow \tau_{k_i}$, $s_i \leftarrow \tau_{k_i+1}$
  retrieve $\tilde{z}_i^{(u_i)}, \tilde{z}_i^{(s_i)}$ from the cached trajectory
end for
predict clean endpoint $\hat{x}_\theta^i \leftarrow \mathrm{Endpoint}\big(f_\theta(\tilde{z}_i^{(u_i)}, u_i, c)\big)$ for all i
consistency kernel mean $\mu_\theta^{u_i \to s_i} \leftarrow \alpha_{s_i} \hat{x}_\theta^i$ ▷ Eq. (6)
per-trajectory stop-gradient regression loss

$$\ell^i \leftarrow \left\| \hat{x}_\theta^i - \mathrm{sg}\left(\hat{x}_\theta^i + \frac{\hat{A}_i \, \alpha_{s_i}}{2\sigma_{s_i}^2} \big(\tilde{z}_i^{(s_i)} - \mu_\theta^{u_i \to s_i} \big) \right) \right\|^2 \qquad \triangleright \text{Eq. (9)}$$

split each ℓ^i along the chunk axis into per-chunk losses ℓ_t^i
compute future participation scores p_t and chunk weights $w_t \propto g_\eta(p_t)$
chunk-weighted CM-GRPO loss $\mathcal{L}_{\text{CM-GRPO}} \leftarrow \dfrac{\sum_{i,t} w_t \ell_t^i}{\sum_{i,t} w_t m_t}$
$\theta \leftarrow \theta - \eta_\theta \nabla_\theta \mathcal{L}_{\text{CM-GRPO}}$

scores each rollout with the composite reward, and converts the endpoint scores into a group-relative advantage following Eq. (9). A single transition $u \to s$ is then sampled along each rollout, and the policy update is realized as the stop-gradient regression on the predicted clean endpoint of the consistency kernel, inheriting the chunk-wise weighting introduced by RAVEN so that supervision remains aligned with the autoregressive horizon. The transition index is restricted to $k_i \in \{1, \dots, K-1\}$ rather than the full set of consistency steps, since the final transition lands at $\tau_K = 0$ with $\sigma_{\tau_K} = 0$, where the kernel collapses to a Dirac delta and the policy log-probability is undefined.

B More Implementation Details

Dataset. Both RAVEN and CM-GRPO are trained exclusively on text prompts drawn from VidProM [128], preprocessed through filtering and large language model extension, following the data protocol of Self Forcing [31]. Ablation experiments that require real video data draw from OpenVidHD-0.4M [129], with all video clips temporally upsampled via RIFE [130] prior to use.

Training Details. Most RAVEN training settings are inherited from Self Forcing [31] and Causal Forcing [36]. We disable weight decay throughout and reduce the two-time-scale update rule (TTUR) [40] ratio between critic and generator updates from 5 to 2. CM-GRPO is instead optimized in a parameter-efficient manner on top of the RAVEN checkpoint, applying LoRA [131] of rank 256 to all linear layers under AdamW [132, 133] with learning rate 5×10^{-6}, zero weight decay, betas $(0.0, 0.999)$ and epsilon 10^{-10}, where each policy update consumes a batch size of 8 paired with a group size of 32. The training-time test framework and chunk-wise loss scaling introduced by RAVEN are carried over unchanged into the CM-GRPO stage, so that the policy update inherits the same alignment between training context and inference-time extrapolation. The Causal Forcing [36] + CM-GRPO entry in Table 1 does not utilize these two designs, so as to isolate the contribution of the policy objective from RAVEN itself. For this entry, the learning rate is lowered to 2×10^{-6} and weight decay is reinstated at 0.01, while the reward weight on the *Dynamic Degree* (DD) dimension is raised from 0.35 to 2.35 to compensate for the weaker motion of the Causal Forcing checkpoint. RAVEN and CM-GRPO consume training budgets of approximately 70 and 170 NVIDIA H200 GPU hours respectively.

Reward Composition. Temporal dynamics are captured through dynamic degree and motion smoothness, where the former takes the top-5% mean of optical flow magnitudes estimated by RAFT [124] across consecutive frame pairs as an index of peak scene motion, and the latter is derived from the reconstruction error incurred when AMT [123] recovers artificially dropped frames from their temporal neighbors, with lower error reflecting greater temporal coherence. At the frame level, the LAION aesthetic predictor [122] applies a linear model on image embeddings to score each frame for compositional appeal and perceptual naturalness, while the multi-scale MUSIQ [121] evaluates low-level technical distortion. Text-video alignment is additionally assessed by VideoReward [118], a reward model built on Qwen2-VL [134] and trained via direct preference optimization [135] on 182K pairwise human preference annotations.

Table 7: Training pipeline decomposition and approximate compute budgets in GPU hours, with wall-clock time $\times$ GPU count in parentheses. RAVEN shares the first two stages with Causal Forcing and replaces only the Self-Forcing DMD stage, and n.r. denotes not reported.

Method	Stage-1 AR Diffusion	Stage-2 ODE Distillation	Stage-3 DMD Training	Stage-4 Online RL
Self Forcing [31]	—	bidirectional (n.r.)	100 (1.5h $\times$ 64)	—
Causal Forcing [36]	n.r.	causal (n.r.)	100 (as Self Forcing)	—
RAVEN + CM-GRPO	as Causal Forcing	as Causal Forcing	**70 (8.7h $\times$ 8)**	170 (21.2h $\times$ 8)

Compute Budget. All training runs on a single node of 8 NVIDIA H200 GPUs. With fully sharded data parallelism and activation checkpointing, the peak memory stays between 75 and 77 GB per GPU for both the 1.3B and 14B models, so the entire pipeline fits any 8-GPU node with at least 80 GB of memory per GPU. Table 7 places our budgets alongside Self Forcing [31] and Causal Forcing [36]. Causal Forcing builds on Self Forcing by adding autoregressive diffusion pretraining and causal ODE distillation, while its DMD stage is exactly the Self-Forcing DMD. RAVEN starts from the same causal ODE initialization and replaces only this shared stage, so it introduces no additional training stage, and its DMD stage is about 30 GPU hours cheaper with a peak memory about 21 GB lower under identical settings. The saving comes from the number of generator passes behind each update. For a 5-second video of 21 latent frames with 3 frames per chunk and 4 denoising steps, Self Forcing runs 23.5 generator passes per iteration on average, namely 17.5 denoising passes and 6 KV-caching passes, whereas RAVEN reuses the self rollout from the fake-score step and calls the generator exactly once on the repacked sequence $\mathcal{I}_u$. Training the 14B model takes 56 hours on the same 8 GPUs.

Table 8: Rollout-horizon scaling of RAVEN at 1.3B on a single H200 GPU (**left**) and generation-seed uncertainty on VBench against Causal Forcing (CF) (**right**).

Horizon	Peak Mem. (GB)	Rollout (s)	Replay (s)	Total (s)
5s	29.29	4.01	4.45	8.46
10s	40.20	8.20	7.84	16.04
20s	65.62	17.98	14.72	32.70
30s	89.25	27.29	21.69	48.98

Seed	Total Score		Dyn. Deg.	
	RAVEN	CF	RAVEN	CF
0	85.15	84.96	2.951	2.669
1	85.29	85.14	2.948	2.665
2	85.15	85.09	2.952	2.664
3	85.28	85.05	2.947	2.669
Mean	**85.22**	85.06	**2.949**	2.666
Std	0.07	0.07	0.002	0.002

Table 9: Validation reward progression of the EMA checkpoint on a fixed prompt set across RL post-training.

Policy Interface	Progress	Text Alignment	Dynamic Degree	Aesthetic Quality	Imaging Quality
RAVEN + EM ($\sigma = 0.8$, $\beta = 0$)	0%	1.403	26.80	0.655	0.677
	25%	1.406	26.72	0.656	0.676
	50%	1.407	26.82	0.655	0.677
	75%	1.408	27.22	0.656	0.678
	100%	1.416	26.94	0.657	0.679
RAVEN + CM-GRPO (Ours)	0%	1.403	26.80	0.655	0.677
	25%	1.413	26.80	0.656	0.678
	50%	1.414	26.48	0.656	0.679
	75%	1.411	26.09	0.658	0.680
	100%	**1.427**	**27.43**	**0.659**	**0.680**

Rollout-Horizon Scaling. The left part of Table 8 measures a single RAVEN iteration at 1.3B on one H200 GPU with a 6-chunk sliding window. Both the per-iteration time and the peak memory grow linearly with the rollout horizon, and training stays within an 80 GB budget up to 20-second rollouts without any memory mechanism. Longer horizons can be reached by pairing the interleaved pass with a bounded-memory cache policy, as discussed in Appendix D.

Reward Progression. Table 9 tracks the EMA checkpoint on a fixed prompt set throughout RL post-training for CM-GRPO and the strongest EM interface without KL regularization. Starting from the same RAVEN checkpoint and the same reward composition, CM-GRPO ends ahead on text alignment, dynamic degree, and aesthetic quality while keeping imaging quality, so the same reward signal is converted into larger improvements once the policy kernel matches the inference sampler.

Seed Uncertainty. The right part of Table 8 re-evaluates RAVEN and Causal Forcing with four generation seeds under identical checkpoints and prompts. RAVEN stays ahead of Causal Forcing on total score for every seed, and its gain in dynamic degree is two orders of magnitude larger than the seed-level standard deviation. The margins of CM-GRPO over RAVEN and over the strongest EM interface in Table 4 are likewise several times this standard deviation.

C User Study

We conduct a user study on 100 long and detailed prompts drawn from the qualitative showcases of the existing baselines, generating 4 samples per prompt for each method. The study covers the four baselines designed for 5-second short video generation, namely CausVid [35], Self Forcing [31], Reward Forcing [33], and Causal Forcing [36]. For each sample pair, an individual user rates a RAVEN clip against its baseline counterpart presented in randomized order along *Quality*, *Semantic*, and *Overall*, following the instructions in Figure 7. Aggregate preference rates are reported in Figure 6, where RAVEN is preferred on every dimension against all four baselines, with a more pronounced lead on *Semantic* than on *Quality* and a clear margin on *Overall*.

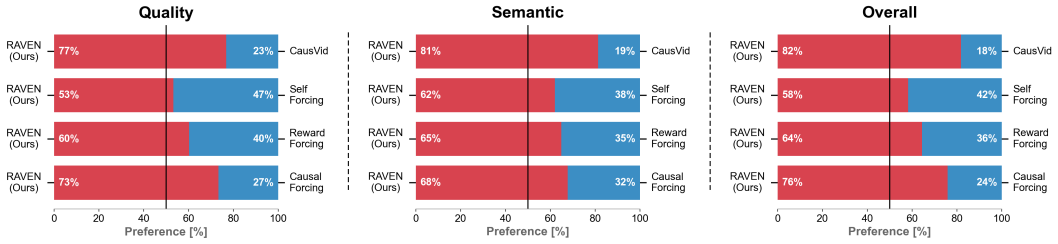


Figure 6: User study preference rates on *Quality*, *Semantic*, and *Overall*.

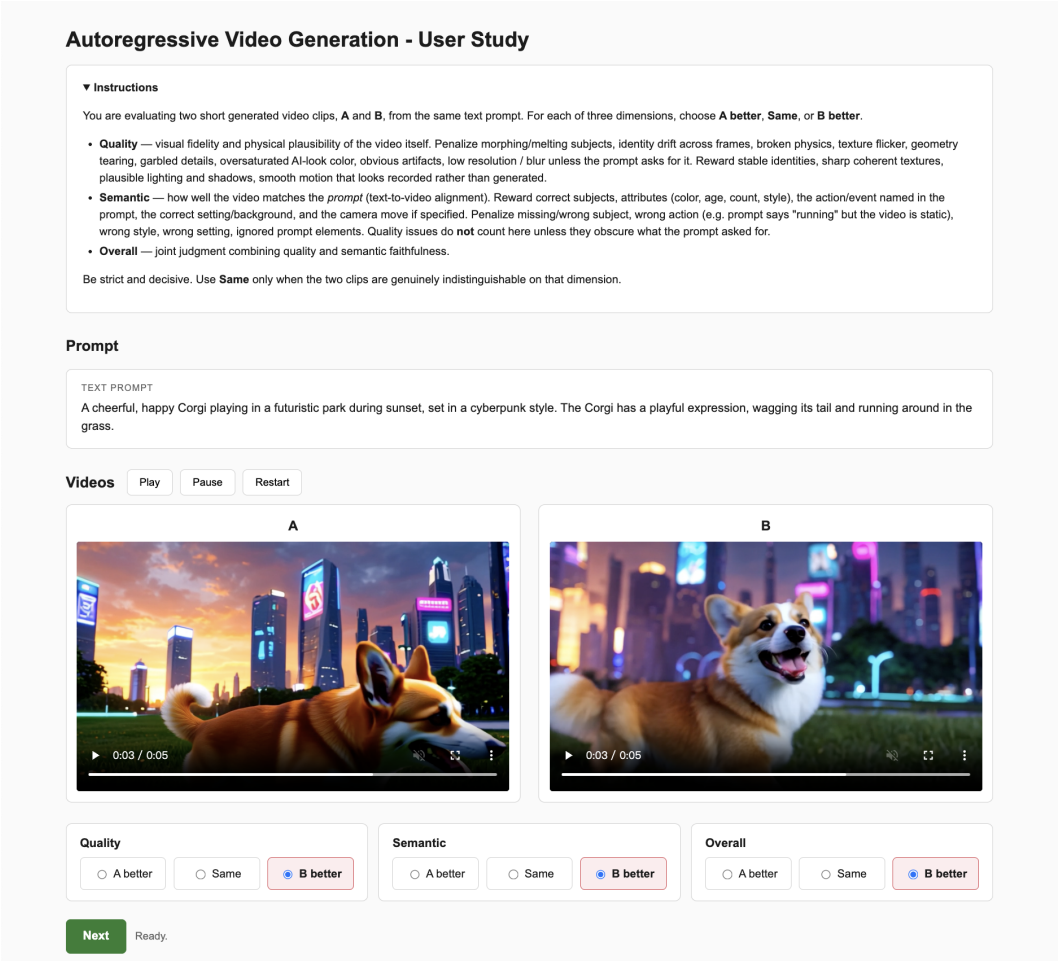


Figure 7: User study instruction screenshot.

Table 10: Additional pairwise user studies following the protocol of Figure 6. Each pair of rows is an independent pairwise comparison, so the two rates in a pair sum to 100%.

Method	Qual. Pref.	Sem. Pref.	Over. Pref.
History supervision under identical weighting (1.3B, Table 2)			
Self Forcing (detached)	48.5%	35.5%	37.0%
RAVEN	**51.5%**	**64.5%**	**63.0%**
Model scale (1.3B and 14B)			
Self Forcing (1.3B)	47.0%	38.0%	42.0%
RAVEN (1.3B)	**53.0%**	**62.0%**	**58.0%**
Self Forcing (14B)	41.0%	42.5%	35.0%
RAVEN (14B)	**59.0%**	**57.5%**	**65.0%**
Policy objective (1.3B, Table 4)			
RAVEN	47.0%	47.7%	46.5%
RAVEN + CM-GRPO	**53.0%**	**52.3%**	**53.5%**
RAVEN + EM ($\sigma = 0.8$, $\beta = 0.004$)	49.0%	49.5%	44.0%
RAVEN + CM-GRPO	**51.0%**	**50.5%**	**56.0%**

Additional User Studies. Table 10 reports further pairwise studies under the same protocol. Against the detached Self Forcing variant of Table 2, which shares the same initialization and chunk-wise loss scaling, RAVEN is clearly preferred on semantic alignment and overall, so raters respond to history supervision rather than to the weighting schedule. Against Self Forcing at both model sizes, the overall preference for RAVEN rises from 1.3B to 14B even though the automatic scores saturate. Raters also prefer RAVEN with CM-GRPO over both RAVEN alone and the strongest EM interface on every dimension, showing that the gains of CM-GRPO are visible to humans and not specific to the reward metrics.

D Discussion

Although RAVEN and CM-GRPO are presented with concrete design choices tailored to causal autoregressive video distillation, both formulations admit broader scope than the setting evaluated in our experiments. The interleaved sequence construction underlying RAVEN currently treats clean chunks as historical context, yet the supervised forward pass does not restrict the form of cached history. Arbitrary representations derived from the rollout can be substituted in its place, including intermediate noisy states from preceding chunks, memory tokens specific to the underlying architecture, and cache management strategies such as sliding windows or attention sinks, provided the same representation is consumed by subsequent chunks during self rollout. These representations can in principle be instantiated inside the supervised forward pass and optimized end-to-end through downstream losses on later noisy states, which would turn history handling into a learnable rather than fixed element of the rollout.

CM-GRPO admits an analogous extension along a different axis. Its policy interface depends only on the conditional Gaussian transition induced by the consistency sampler and is independent of the autoregressive structure of the generator on which it is applied. The same objective therefore applies to any few-step generator that draws samples through a stochastic consistency step, encompassing bidirectional video models as well as generators in other modalities that have been distilled through consistency training or distillation. Collectively, RAVEN constitutes a general training-time test interface for autoregressive few-step distillation, while CM-GRPO constitutes a general policy optimization interface for consistency generators, indicating that both contributions extend beyond the specific setting studied here.

Memory-Based and Compressed History. RAVEN only requires that subsequent chunks in the supervised pass consume the same history representation the self rollout produced, so cache policies used at inference can be brought inside the supervised graph. For an attention sink with a sliding window, the sink chunk is never evicted in the repacked sequence, every later noisy chunk attends to it together with its local window, and the sink representation is trained end to end. A compressed-history design that keeps recent chunks at full resolution and folds older chunks into a compact memory needs one more step, because the same early chunk is seen in full by nearby noisy chunks and in

compressed form by distant ones. Suppose chunks 1 to 3 are compressed into a memory token c_1 once chunk 4 is generated. The repacked sequence then becomes

$$\mathcal{I}_u^{\mathrm{comp}} = \big(\hat{z}_1^{(u)}, \hat{x}_1, \hat{z}_2^{(u)}, \hat{x}_2, \hat{z}_3^{(u)}, \hat{x}_3, \hat{z}_4^{(u)}, c_1, \hat{x}_4, \hat{z}_5^{(u)}, \hat{x}_5, \hat{z}_6^{(u)}\big), \tag{11}$$

where the attention mask routes $\hat{z}_2^{(u)}$ to $\hat{z}_4^{(u)}$ to the uncompressed prefix and routes $\hat{z}_5^{(u)}$ and $\hat{z}_6^{(u)}$ to c_1 together with $\hat{x}_4$ and $\hat{x}_5$. The construction of c_1 thereby enters the supervision scope. If compressing away a detail hurts the denoising of $\hat{z}_5^{(u)}$ or $\hat{z}_6^{(u)}$, the resulting gradient flows through c_1 into the compressor, which learns to keep what later generation depends on.

Limitations. Without a bounded-memory mechanism, both the self rollout and the interleaved generator pass grow linearly with the number of chunks, since the repacked sequence holds both clean endpoints and noisy states and is about 1.86 times the latent length at seven chunks. Training therefore fits an 80 GB GPU up to 20-second rollouts, and longer horizons require a sliding-window, sink, or compressed-memory policy inside the interleaved pass. A learnable compressor must in addition place both full and compressed views of the same chunk in the packed sequence with view-specific attention routing, which increases the sequence length and mask complexity and is left for future work.

E Prompts

We list the text prompts behind the qualitative comparisons in the main text, indexed by their position within each figure grid.

Figure 3 (qualitative comparison).

- *Top-left.* A cheerful, happy Corgi playing in a futuristic park during sunset, set in a cyberpunk style. The Corgi has a playful expression, wagging its tail and running around in the grass. The park is illuminated by neon lights and surrounded by towering skyscrapers with holographic advertisements flashing on their surfaces. The sky is a blend of orange and purple hues, creating a striking contrast against the dark cityscape. The Corgi is in the foreground, while the vibrant city lights and buildings create a dynamic background. The scene captures the essence of a cyberpunk world with natural elements intertwined. Medium close-up shot focusing on the Corgi's joyful playfulness.
- *Top-right.* A majestic African lion with golden fur and piercing green eyes grabs a small antelope tightly in its powerful jaws. The lion's mane flows gracefully as it crouches on all fours, muscles tense and alert. After a moment of contemplation, the lion releases the antelope, which quickly runs away. The scene takes place in a dense savanna with tall grass and scattered acacia trees. The lion remains still for a few moments before turning its gaze back to the fleeing antelope. Wide shot, with a focus on the lion's expressions and movements.
- *Bottom-left.* A cheerful woman with a warm smile, standing outdoors under a clear blue sky. Her hair is flowing freely as she faces the wind, creating a gentle, swirling wind effect around her. She is wearing a light, flowy dress that moves gracefully with the breeze. The background includes green trees and blooming flowers, adding a vibrant and lively atmosphere. The scene captures a close-up of the woman, emphasizing her joyful expression and the dynamic wind movement around her.
- *Bottom-right.* On a rainy day, a young boy with tousled brown hair is sprinting away from the rain, his clothes slightly damp and clinging to his body. His face is animated with excitement and joy as he runs barefoot through puddles, splashing water everywhere. The sky is overcast, with heavy raindrops falling steadily. In the background, blurred figures of people walking with umbrellas can be seen. The scene is captured in a medium shot, focusing on the boy's dynamic movements and expressions.

Figure 4 (qualitative ablation).

- *Left.* A documentary-style nature photography shot from a camera truck moving to the left, capturing a crab quickly scurrying into its burrow. The crab has a hard, greenish-brown shell and long claws, moving with determined speed across the sandy ground. Its body is slightly arched as it burrows into the sand, leaving a small trail behind. The background shows a shallow beach with scattered rocks and seashells, and the horizon features a gentle curve of the coastline. The

photo has a natural and realistic texture, emphasizing the crab's natural movement and the texture of the sand. A close-up shot from a slightly elevated angle.

- *Middle.* A dynamic action shot of a surfer accelerating on a powerful wave, carving through the water with grace and agility. The surfer, with a tanned complexion and muscular build, rides the wave with one hand gripping the board while the other extends outwards for balance. The water splashes behind, creating a foamy trail, and the sun casts a golden glow over the scene. The background features a clear blue ocean and distant white-capped waves, with a few seagulls flying overhead. The surfer's expression is one of exhilaration and focus. A mid-shot from a low-angle perspective capturing the surfer's motion and the wave's power.
- *Right.* A scenic photograph capturing the moment a steam train departs from the Glenfinnan Viaduct, a historic railway bridge in Scotland. The train moves gracefully over the arch-covered viaduct, its smoke billowing into the air. The landscape is lush with greenery, and towering rocky mountains frame the scene, creating a picturesque backdrop. The sky is a clear, bright blue with the sun shining down, casting a warm glow on the train and the surrounding scenery. The viaduct itself is a striking feature, with intricate ironwork and a verdant setting. The photo has a classic, nostalgic feel, emphasizing the natural beauty and historical charm of the location. A wide-angle shot from a slightly elevated angle, capturing both the train and the expansive landscape.

F Broader Impacts

RAVEN and CM-GRPO advance the practical viability of real-time autoregressive video generation, which carries both positive and negative societal implications. On the positive side, real-time low-latency video synthesis can support creative tools, education, accessibility applications such as sign language generation [136], and interactive simulation for embodied agents [137], while reducing the compute footprint of video generation pipelines and lowering the barrier for downstream research. On the negative side, more capable and more accessible video generators raise familiar misuse concerns, including the production of misleading or non-consensual synthetic media, identity impersonation, and content that could be used to spread misinformation. These risks are inherited from the underlying generative models rather than introduced by our contributions, but the gain in efficiency does broaden the population of users who can produce such content. We encourage downstream applications to combine model release with provenance tooling, watermarking, and content moderation, and to follow community norms for the responsible release of generative video models.

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 describe the two contributions (RAVEN as a training-time test framework for autoregressive few-step distillation and CM-GRPO as a policy optimization objective for consistency generators) at the same scope demonstrated by the experimental results.

   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: Appendix D discusses the empirical scope of our results, the linear growth of compute and memory with the rollout horizon, and the added complexity of integrating learnable history compressors into the interleaved training sequence.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: The paper does not include formal theorems or proofs; the contributions are empirical training frameworks and a policy optimization objective.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: The methodology section formalizes RAVEN and CM-GRPO, Appendix A provides full per-iteration pseudocode for both, and Appendix B reports datasets, optimizer settings, learning rates, LoRA configuration, reward composition, and compute budgets.

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: We publicly release our training and evaluation code as well as the distilled RAVEN and CM-GRPO model checkpoints via our project page at `https://yanzuo.lu/raven` (code repository at `https://github.com/mvp-ai-lab/RAVEN` and model weights at `https://huggingface.co/mvp-lab/RAVEN`).

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The main text and Appendix B report datasets, training source inheritance from Self Forcing and Causal Forcing, optimizer choice, learning rates, weight decay, betas, epsilon, batch sizes, group size, LoRA rank, reward composition, and the variant-specific overrides used in Table 1.

   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: [Yes]

   Justification: Appendix B reports the mean and standard deviation over four generation seeds for RAVEN and Causal Forcing under the full VBench protocol, and Appendix C reports pairwise human preference studies for every key comparison.

   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: Section 4.3 and Appendix B report the hardware, wall-clock time, GPU hours, and peak memory of every training stage at 1.3B and 14B, compare these budgets with prior baselines, and measure how time and memory scale with the rollout horizon.

   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 the NeurIPS Code of Ethics and our research, including the user study, conforms to its requirements, with anonymity preserved throughout the submission.

   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: Appendix F discusses both the positive applications and the misuse risks of more efficient and accessible video generation, and points to common mitigation directions such as provenance tooling and watermarking.

Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This submission does not release model checkpoints or scraped datasets, and the contributions are training procedures applied on top of publicly available models, so no additional safeguards are introduced at this stage.

    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: All existing assets used for training, evaluation, and reward modeling, including Wan2.1, VidProM, OpenVidHD-0.4M, RIFE, VBench, RAFT, AMT, MUSIQ, the LAION aesthetic predictor, VideoReward, UnifiedReward-32B, and Qwen2-VL, are cited at their original sources and used in accordance with their respective licenses.

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: [Yes]

    Justification: Our released code (`https://github.com/mvp-ai-lab/RAVEN`) and distilled model checkpoints (`https://huggingface.co/mvp-lab/RAVEN`) include full documentation on setup, configuration, training, evaluation, and licensing.

    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: [Yes]

    Justification: The user study setup, the rating dimensions, and a screenshot of the rating interface with full instructions are included in Appendix C; raters are unpaid volunteers from our colleagues, so no compensation arrangements apply.

    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 user study consists of voluntary preference ratings on short generated video clips by colleagues, posing no foreseeable risk to participants beyond ordinary media viewing, and accordingly does not fall under the IRB review requirements at our institution.

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: LLMs are not part of the core methodology of RAVEN or CM-GRPO; their only role in our pipeline is the standard prompt-extension preprocessing of VidProM that we inherit from Self Forcing's data protocol, as noted in Appendix B.

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.