LatSearch: Latent Reward-Guided Search for Faster Inference-Time Scaling in Video Diffusion

Zengqun Zhao[1], Ziquan Liu[1], Yu Cao[1], Shaogang Gong[1], Zhensong Zhang[3], Jifei Song[3], Jiankang Deng[2], Ioannis Patras[1]

[1]Queen Mary University of London, [2]Imperial College London, [3]Huawei R&D UK
https://zengqunzhao.github.io/LatSearch

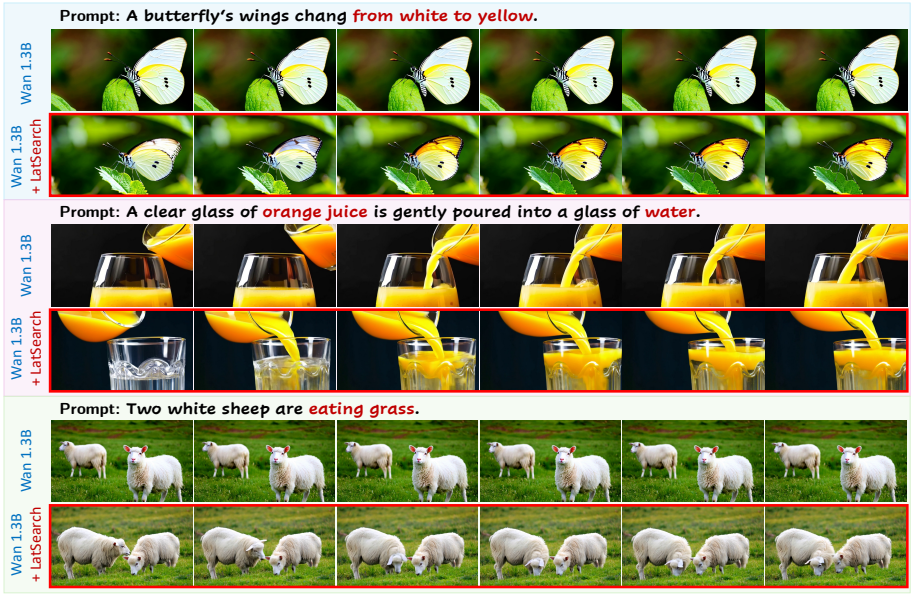

Fig. 1: Text-to-video generations, comparing a vanilla model with LatSearch, a novel faster inference-time scaling method in video generation. LatSearch significantly improves sample quality by leveraging latent reward-guided computation allocation during inference, enabling early evaluation of noisy latents and the selection of credible candidates along the diffusion trajectory.

Abstract. The recent success of inference-time scaling in large language models has inspired similar explorations in video diffusion. In particular, motivated by the existence of "golden noise" that enhances video quality, prior work has attempted to improve inference by optimising or searching for better initial noise. However, these approaches have notable limitations: they either rely on priors imposed at the beginning of noise sampling or on rewards evaluated only on the denoised and decoded videos. This leads to error accumulation, delayed and sparse reward signals, and prohibitive computational cost, which prevents the use of stronger search algorithms. Crucially, stronger search algorithms are precisely what could unlock substantial gains in controllability, sample efficiency and generation quality for video diffusion, provided their computational cost can be

reduced. To fill in this gap, we enable efficient inference-time scaling for video diffusion through latent reward guidance, which provides intermediate, informative and efficient feedback along the denoising trajectory. We introduce a latent reward model that scores partially denoised latents at arbitrary timesteps with respect to visual quality, motion quality, and text alignment. Building on this model, we propose LatSearch, a novel inference-time search mechanism that performs Reward-Guided Resampling and Pruning (RGRP). In the resampling stage, candidates are sampled according to reward-normalised probabilities to reduce over-reliance on the reward model. In the pruning stage, applied at the final scheduled step, only the candidate with the highest cumulative reward is retained, improving both quality and efficiency. We evaluate LatSearch on the VBench-2.0 benchmark and demonstrate that it consistently improves video generation across multiple evaluation dimensions compared to the baseline Wan2.1 model. Compared with the state-of-the-art, our approach achieves comparable or better quality while reducing runtime by up to 79%.

Keywords: Video generation · Diffusion models · Latent reward models

1 Introduction

Given the wide range of applications of video generation, such as video editing [25], customisation [26], image animation [13], and world modelling [1], there has been a growing interest in transferring the success of inference-time scaling observed in large language models (LLMs) to diffusion-based video generation models [30,34]. A natural way for better fidelity is to increase the number of denoising steps, which directly improves sample fidelity. However, recent research has shown that inference-time scaling extends further than simply increasing denoising steps [34]. Several studies have demonstrated the effectiveness of so-called "golden noise", specific initial noise realisations that reliably lead to higher-quality generations, highlighting the significant role of noise initialisation in the final generation quality [2, 27, 42, 63]. Consequently, many works now attempt to allocate additional computation at inference time to improve the fidelity and consistency of generated videos [20,34,37,56].

A key problem in inference-time scaling for video diffusion lies in the inability to evaluate intermediate latents reliably. Lacking such evaluations prohibits a model from supporting more flexible strategies, such as early stopping for efficiency, resulting in errors introduced at the initial stage being accumulated throughout the long denoising trajectory. Existing methods, however, largely overlook this problem. Noise optimisation techniques bias the initialisation by injecting noise into a reference video, applying temporal warping or optical flow, or fusing frequency components of denoised latents [7, 9, 54, 59, 60]. However, once the trajectory begins, they lack mechanisms to monitor and correct intermediate states. Noise search methods instead generate multiple candidates and select the best based on fully decoded videos. Such models leverage strategies such as Best-of-N sampling, beam search, evolutionary algorithms, or path

search [20, 34, 37, 46, 56]. These methods typically depend on reward models or verifiers, which are chosen from standard metrics (FID, IS, DINO, CLIP), or video-specific reward functions [31]. More recently, uncertainty measures derived from attention maps have also been considered [27]. However, current noise search methods operate only on final outputs, causing them to incur high computational cost from full video decoding and suffer from reward delay [29], limiting their usefulness in guiding generation.

To address this limitation, we propose LatSearch, a faster and better inference time scaling method that integrates a latent reward model into video diffusion. Unlike conventional verifiers that operate only on final decoded videos, our reward model evaluates partially denoised latents at arbitrary timesteps, providing intermediate feedback on generation progress to facilitate efficient inference-time search. By introducing process-level supervision, LatSearch can identify and prune low-quality candidates early, thereby reducing unnecessary denoising steps. This not only mitigates error accumulation and reward delay but also avoids the heavy cost of repeatedly decoding full videos, making a more efficient and effective search procedure possible. To enable inference-time search guided by intermediate latents, we propose a latent reward model that evaluates partially denoised latents at arbitrary timesteps with respect to visual quality, motion quality, and text alignment. This model directly guides the search process by scoring intermediate latents, enabling more fine-grained candidate selection than final-video evaluation. For training, we construct a dataset of (prompt, latent, timestep, video score, latent similarity) tuples, where latent similarity measures the correspondence between an intermediate latent and the final clean latent. The model is optimised with both a regression loss and a latent preference loss to improve accuracy and robustness in intermediate reward estimation. Building on this model, LatSearch introduces Reward-Guided Resampling and Pruning (RGRP) to refine noise candidates during generation. The resampling stage is inspired by importance sampling: instead of deterministically picking the highest-reward candidate, we sample candidates according to reward-normalised probabilities. This prevents over-reliance on the reward model, which may be imperfect. The pruning stage, applied at the final scheduled timestep, selects the single candidate with the highest cumulative reward across timesteps, reducing redundancy and significantly lowering the computational cost of the remaining denoising process.

In summary, the main contributions of this work are as follows:

- We propose a reward model that evaluates partially denoised latents at arbitrary timesteps, providing process-level supervision on visual quality, motion quality, and text alignment. Since intermediate latents lack explicit semantics, we introduce a similarity-based grounding strategy and combine regression with preference losses to make latent-level reward estimation feasible.
- Building on this reward model, we design an inference-time scaling algorithm that incorporates intermediate supervision directly into the denoising trajectory. Candidates are probabilistically resampled according to reward-normalised weights, duplicates are removed via uniqueness pruning, and cu-

mulative rewards are used for final selection, making latent reward actionable for efficient search.
– On VBench2.0, LatSearch consistently improves video generation across creativity, commonsense, controllability, human fidelity, and physics. Compared with state-of-the-art inference-time scaling methods, our approach achieves comparable or better quality while reducing runtime by up to 79%.

2 Related Work

We review related work in two areas most relevant to our study: video generation methods and inference-time scaling for diffusion models.

Video Generation. Early studies on video generation explored diverse paradigms, including VAEs [4, 47], GANs [6, 24], and autoregressive models [14, 18]. More recently, diffusion-based approaches have become the dominant paradigm [13,19,57], achieving superior visual fidelity and scalability by extending the success of image diffusion models. Progress in video diffusion has generally followed two directions: (i) foundational models, which adapt image diffusion architectures to the temporal domain, and (ii) enhancements, which improve fidelity, efficiency, and controllability. In the text-to-video (T2V) setting, early models extended U-Net–based image diffusion backbones with temporal modules, as in Stable Video Diffusion (SVD) [5]. Subsequent works introduced improved training strategies for temporal coherence and motion quality, exemplified by ModelScope [52], VideoCrafter [10], and AnimateDiff [19]. More recently, Diffusion Transformers (DiTs) [40] have emerged as stronger backbones, offering improved scalability and modelling capacity. At the same time, training objectives have evolved beyond conventional denoising losses toward flow-based formulations, enabling more efficient and stable optimisation. Works such as UViT [3] and Gentron [11] first demonstrated the feasibility of Transformer-only backbones, inspiring large-scale systems like Hunyuan Video [28], CogVideoX [57], and Wan [51]. Building on these foundation models, subsequent research has explored generating longer videos [35,39,43], improving temporal coherence [33,36,41], leveraging human feedback [22, 31, 58, 64], enhancing controllability [12, 55], and improving generation efficiency [50]. In our work, we build on the state-of-the-art Wan model as the foundation for video generation, and investigate inference-time scaling through latent reward guidance. While prior work has explored latent reward models for few-shot video generation [16], they are limited to evaluating the final denoised latent. In contrast, our approach evaluates intermediate states across denoising timesteps, enabling more fine-grained and effective guidance.

Inference-Time Scaling for Diffusion Models. There is growing interest in inference-time scaling for diffusion models, inspired by its success in large language models (LLMs) [30, 34]. A straightforward scaling strategy is to increase the number of denoising steps, which generally improves sample fidelity. However, recent studies show that inference-time scaling extends far beyond this [34]. In particular, the choice of initial noise has been identified as a key factor in generation quality, with the notion of "golden noise" underscoring its importance [2, 27, 42, 63]. As a result, many methods now dedicate additional

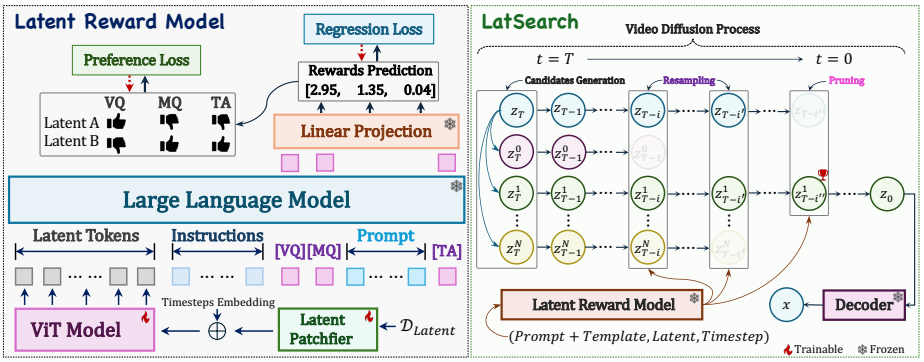

Fig. 2: An overview of a latent reward model (left) and the proposed latent reward-guided inference-time search method, LatSearch (right). On the left, input latent tokens are patchified, fused with timestep embeddings, and projected by a ViT encoder. Together with instruction tokens, text prompts, and special query tokens ([VQ], [MQ], [TA]), these form the input to a large language model. The model is trained using a combination of regression and preference losses. On the right, LatSearch maintains multiple candidate trajectories during a diffusion process. Candidates are periodically scored by the latent reward model, resampled with uniqueness to encourage diversity, and finally pruned based on cumulative rewards before decoding into the final video.

computation at inference time to either optimise the initial noise or search for better noise samples [20, 34, 37, 56]. Noise optimisation approaches inject priors into noise initialisation, for example by adding noise to a reference video [60], warping noise via temporal correlation or optical flow [7, 9], or fusing frequency components of denoised latents [54, 59]. While fusion-based methods avoid reliance on external inputs, they often require more iterations. Noise search approaches instead generate multiple candidates and evaluate the resulting videos. Techniques include Best-of-N sampling [46], beam search [37, 56], evolutionary search [20], and search-over-path strategies [34]. Candidate evaluation typically relies on reward models or verifiers, ranging from traditional metrics such as FID [21], IS [45], DINO [8], and CLIP [44], to video-specific reward models designed for temporal quality assessment [31]. Beyond explicit search, recent work has also explored estimating noise quality directly from model attention maps using uncertainty-based measures [27]. Existing inference-time scaling methods optimise or search over initial noise and evaluate only final videos, lacking intermediate guidance. In contrast, we propose a latent reward model that assesses partially denoised latents at arbitrary timesteps, providing fine-grained feedback on visual quality, motion, and text alignment.

3 Method

Our approach, LatSearch, integrates a latent reward model with a reward-guided search mechanism to scale video diffusion at inference time. In this section, we first review the preliminaries of video diffusion models, then introduce a latent reward model that provides intermediate evaluations throughout the denoising process, and finally describe how these signals are incorporated into a reward-guided search strategy for efficient and high-quality video generation.

3.1 Preliminaries

Latent text-to-video diffusion models encode an F-frame clean video $\{\boldsymbol{x}^{(i)}\}_{i=1}^{N} \in \mathbb{R}^{F\times C\times H\times W}$ into latent representations $\{\boldsymbol{z}_0^{(i)}\}_{i=1}^{N}$ using an encoder $\mathcal{E}$, where C, H, W denote the channel, height, and width of each frame. For convenience, we denote $\boldsymbol{z}_0 = \{\boldsymbol{z}_0^{(i)}\}_{i=1}^{N} \sim p_0(\boldsymbol{z})$. The forward diffusion process [15] gradually perturbs $\boldsymbol{z}_0$ into a noised latent $\boldsymbol{z}_t$ according to

$$q(\boldsymbol{z}_t \,|\, \boldsymbol{z}_0) = \mathcal{N}\big(\boldsymbol{z}_t; \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{z}_0,\ \bar{\alpha}_t\mathbf{I}\big), \tag{1}$$

where $\bar{\alpha}_t$ is the noise schedule coefficient at timestep t.

In text-to-video generation, a prompt p is encoded into a condition $\boldsymbol{c} = \mathcal{E}_{\text{text}}(p)$, which guides the denoising process. The training objective minimises the mean squared error between the true noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and the predicted noise:

$$\mathbb{E}_{\boldsymbol{z}_0,\boldsymbol{\epsilon},t,c}\Big[\ \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\boldsymbol{z}, t, \boldsymbol{c})\|^2\ \Big], \tag{2}$$

where $\boldsymbol{\epsilon}_\theta(z_t, t, \boldsymbol{c})$ is the noise predictor parameterized by θ.

After training, we generate videos by solving the reverse diffusion ODE [49] using the UniPC sampler [61], a second-order predictor–corrector framework designed for fast and accurate sampling. The ODE is expressed as

$$\frac{d\boldsymbol{z}_t}{dt} = f_\theta(\boldsymbol{z}_t, t, \boldsymbol{c}), \tag{3}$$

where f_θ is derived from $\boldsymbol{\epsilon}_\theta$ under the probability-flow formulation. To strengthen text guidance, we apply classifier-free guidance (CFG) [23]:

$$\boldsymbol{\epsilon}_\theta^{w}(\boldsymbol{z}_t, t, \boldsymbol{c}) = \boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t, \varnothing) + w\big[\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t, \boldsymbol{c}) - \boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t, t, \varnothing)\big], \tag{4}$$

where $w \in \mathbb{R}_{\geq 0}$ is the guidance scale and $\varnothing$ denotes the null text prompt.

At each denoising step $t_s \to t_{s-1}$ with step size h_s, UniPC applies a second-order update of the form

$$\boldsymbol{z}_{t_{s-1}} \ = \ \boldsymbol{z}_{t_s} + \tfrac{h_s}{2}\Big(f_\theta(\boldsymbol{z}_{t_s}, t_s, \boldsymbol{c}) + f_\theta(\tilde{\boldsymbol{z}}_{t_{s-1}}, t_{s-1}, \boldsymbol{c})\Big), \tag{5}$$

where $\tilde{\boldsymbol{z}}_{t_{s-1}} = \boldsymbol{z}_{t_s} + h_s f_\theta(\boldsymbol{z}_{t_s}, t_s, \boldsymbol{c})$ serves as a predictor. Finally, the terminal latent $\boldsymbol{z}_0$ is decoded by $\mathcal{D}$ into an N-frame RGB video, $\{\boldsymbol{x}^{(i)}\}_{i=1}^{N} = \mathcal{D}(\boldsymbol{z}_0)$.

3.2 Latent Reward Model

To enable intermediate evaluations during video diffusion, we design a latent reward model that can assign quality scores to partially denoised latents without decoding to the video space. We first describe the construction of the training dataset, then detail the architecture of the reward model, and finally present the objective function used to optimise it.

Latent Reward Data Construction. Most reward assessors operate on rendered videos and return video-level scores, making it nontrivial to supervise rewards directly on latent representations. Concretely, given a prompt p and the

final denoised ("clear") latent $\boldsymbol{z}_0$, the video-level reward vector is obtained on the decoded video

$$\boldsymbol{r} \;=\; \big(r^{\mathrm{VQ}},\, r^{\mathrm{MQ}},\, r^{\mathrm{TA}}\big)^{\top} \;=\; \mathcal{R}\big(\mathcal{D}(\boldsymbol{z}_0),\, p\big), \tag{6}$$

where $\mathcal{D}$ is the decoder, p is the prompt, and $\mathcal{R}$ denotes external verifiers or human annotations for visual quality (VQ), motion quality (MQ), and text alignment (TA) [31]. However, our reward model must evaluate intermediate latents $\boldsymbol{z}_t$ extracted along the denoising trajectory. Since direct latent-level ground truth is unavailable, we propose a similarity-grounded credit assignment. Specifically, we ground video-level rewards to intermediate latents by measuring how much an intermediate latent $\boldsymbol{z}_t$ "contributes" to the final clear latent $\boldsymbol{z}_0$ via its similarity to the clear latent. We define a cosine-based similarity, rescaled to $[0, 1]$:

$$s_t \;=\; \frac{1}{2}\left(1 + \frac{\langle \boldsymbol{z}_t,\, \boldsymbol{z}_0 \rangle}{\|\boldsymbol{z}_t\|_2\, \|\boldsymbol{z}_0\|_2}\right) \;\in\; [0, 1]. \tag{7}$$

The similarity s_t quantifies how close $\boldsymbol{z}_t$ is to $\boldsymbol{z}_0$ in a task-relevant representation. Finally, we assign latent-level targets by crediting each dimension of the video-level reward proportionally to s_t:

$$\tilde{\boldsymbol{r}}_t \;=\; s_t \cdot \boldsymbol{r} \;=\; \big(s_t\, r^{\mathrm{VQ}},\; s_t\, r^{\mathrm{MQ}},\; s_t\, r^{\mathrm{TA}}\big)^{\top}. \tag{8}$$

This yields the latent reward dataset

$$\mathcal{D}_{\mathrm{latent}} \;=\; \big\{\, (\boldsymbol{z}_t,\, p,\, \tilde{\boldsymbol{r}}_t,\, t) \;:\; t \in \mathcal{T} \,\big\}, \tag{9}$$

where $\mathcal{T}$ is the set of sampled timesteps.

Model Architecture. The reward model takes three inputs: a latent representation $\boldsymbol{z}_t \in \mathbb{R}^{F/4 \times C' \times H/8 \times W/8}$ (where F, C', H, and W denotes video frames number, latent channel, video height, and video width), the denoising step t, and the text prompt p. These inputs are unified into a token sequence and processed by a transformer-based backbone. *Latent tokens:* The intermediate latent tensor $\boldsymbol{z}_t$ is patchified by a lightweight 3D convolutional encoder, which partitions the spatiotemporal volume into non-overlapping blocks and projects them into embedding vectors. This yields a sequence of video tokens that capture both spatial appearance and temporal motion. *Step embedding:* The denoising step t is mapped to a learnable embedding vector $\mathbf{e}_t$ through an embedding layer. This embedding is concatenated with the token sequence to provide the model with explicit temporal information about the diffusion process. *Prompt tokens:* The text prompt p is formatted using an instruction-style template designed for reward modeling [31], and tokenized into instruction tokens. Special query tokens [VQ], [MQ], and [TA] are appended to the sequence to request predictions for visual quality, motion quality, and text–video alignment, respectively. The overall structure is illustrated on the left of Figure 2.

The unified token sequence is then processed by a transformer-based backbone, which outputs hidden states for the query tokens. Finally, the hidden states

corresponding to the three reward query tokens are extracted and projected by a linear layer to obtain scalar scores:

$$\hat{\boldsymbol{r}} = \mathrm{Linear}\Big(h^{[\mathrm{VQ}]},\, h^{[\mathrm{MQ}]},\, h^{[\mathrm{TA}]}\Big), \tag{10}$$

where $h^{[\cdot]}$ denotes the hidden state of each query token. This yields the predicted rewards $(\hat{r}^{\mathrm{VQ}},\, \hat{r}^{\mathrm{MQ}},\, \hat{r}^{\mathrm{TA}})$.

Training Objective. A latent reward model R_ψ is optimised with a combination of a regression loss and a preference loss, showing in *Appendix Algorithm 1.*

Given the latent tensor $\boldsymbol{z}_t$, prompt p, and denoising step t, the model predicts reward scores $\hat{\boldsymbol{r}} = R_\psi(\boldsymbol{z}_t, p, t)$ across three dimensions. Since the latent reward dataset $\mathcal{D}_{\mathrm{latent}} = \{(\boldsymbol{z}_t, p, \tilde{\boldsymbol{r}}_t, t)\}$ already incorporates the similarity weighting, the regression target for each dimension is $\tilde{\boldsymbol{r}}_t^d$, where $d \in \{\mathrm{VQ}, \mathrm{MQ}, \mathrm{TA}\}$. The regression loss is therefore

$$\mathcal{L}_{\mathrm{reg}}^d = \big\|\hat{\boldsymbol{r}}^d - \tilde{\boldsymbol{r}}_t^d\big\|_2^2. \tag{11}$$

While regression provides absolute supervision, it does not enforce relative ordering among candidates. Inspired by reinforcement learning from human feedback (RLHF) [38], we introduce a preference loss that encourages the model to predict higher scores for better samples. For each pair (i, j) in a minibatch, we define

$$\Delta\hat{\boldsymbol{r}}_{ij}^d = \hat{\boldsymbol{r}}_i^d - \hat{\boldsymbol{r}}_j^d, \qquad \boldsymbol{y}_{ij}^d = \mathbb{I}[\boldsymbol{r}_i^d > \boldsymbol{r}_j^d], \tag{12}$$

where $\boldsymbol{y}_{ij}^d$ denotes the ground-truth preference label for pair (i, j) in dimension d. The preference loss is then formulated as:

$$\mathcal{L}_{\mathrm{pref}}^d = \frac{1}{|\mathcal{P}_d|} \sum_{(i,j)\in\mathcal{P}_d} \log\big(1 + \exp\big(-(2\boldsymbol{y}_{ij}^d - 1)\,\Delta\hat{\boldsymbol{r}}_{ij}^d\big)\big)\,, \tag{13}$$

where $\mathcal{P}_d$ is the set of preference pairs. This is equivalent to applying binary cross-entropy on pairwise score differences.

The final loss is a weighted sum over dimensions d and both objectives:

$$\mathcal{L} = \sum_{d\in\{\mathrm{VQ},\mathrm{MQ},\mathrm{TA}\}} \big(\lambda_{\mathrm{reg}}^d\, \mathcal{L}_{\mathrm{reg}}^d + \lambda_{\mathrm{pref}}^d\, \mathcal{L}_{\mathrm{pref}}^d\big). \tag{14}$$

We optimise the reward model parameters ψ using stochastic gradient descent. Regression loss anchors the predictions to absolute reward magnitudes, while preference loss shapes the reward landscape to preserve relative orderings, analogous to the role of preference modelling in RLHF.

3.3 Latent Search with Reward-Guided Resampling and Pruning

We consider inference-time search as an importance-sampling-inspired procedure on latent trajectories. Standard samplers such as DDIM [48] or UniPC [61] follow a single trajectory, which may fall into suboptimal modes. To improve robustness, we maintain a set of N candidate trajectories, inspired by sequential Monte

Carlo (SMC) methods [17], and iteratively refine this set during denoising. The algorithm of the proposed latent search method is in *Appendix Algorithm 2.*

Candidate generation. At the initial timestep T, we generate candidates by perturbing a base Gaussian noise $\boldsymbol{z}_T^{(0)}$ with isotropic perturbations $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$:

$$\boldsymbol{z}_T^{(i)} \;=\; \sqrt{1-\eta^2}\,\boldsymbol{z}_T^{(0)} + \eta\,\boldsymbol{\epsilon}_i, \qquad i = 1, \ldots, N, \tag{15}$$

where η controls the candidate diversity. Each candidate is evolved independently with a diffusion sampler.

Reward-guided resampling with uniqueness. Let $\mathcal{Z}_t = \{(\boldsymbol{z}_t^{(i)}, \sigma_i)\}_{i=1}^N$ be the candidate set at step t, where σ_i is the seed identity of candidate i (determined by its initial noise). At scoring steps $t \in \mathcal{S}$, where $\mathcal{S}$ is the search schedule, we obtain rewards $r_i^{(t)} = R_\psi(\boldsymbol{z}_t^{(i)}, p, t)$ and convert them to normalized weights

$$\pi_i^{(t)} \;=\; \frac{\exp(\tau\, r_i^{(t)})}{\sum_{k=1}^N \exp(\tau\, r_k^{(t)})}, \qquad \sum_i \pi_i^{(t)} = 1, \tag{16}$$

where $\tau > 0$ is a temperature.

We then draw $\boldsymbol{n}^{(t)} \sim \mathrm{Multinomial}\big(N; \pi_1^{(t)}, \ldots, \pi_N^{(t)}\big)$ with replacement, and keep the unique seeds only:

$$\mathcal{I}^{(t)} \;=\; \mathrm{supp}\big(\boldsymbol{n}^{(t)}\big) \;=\; \{\, i \mid n_i^{(t)} > 0 \,\}, \qquad \mathcal{Z}_t^+ \;=\; \big\{\, (\boldsymbol{z}_t^{(i)}, \sigma_i) \,:\, i \in \mathcal{I}^{(t)} \big\}. \tag{17}$$

The uniqueness operator, $\mathrm{supp}(\cdot)$, removes multiplicities (duplicates of the same seed) and thus avoids wasting compute on identical trajectories, in which the survival probability of candidate i after uniqueness is $1 - \big(1 - \pi_i^{(t)}\big)^N$. This increases monotonically with its weight $\pi_i^{(t)}$.

Between two scoring steps, there are many intermediate denoising updates, during which candidates evolve independently via the diffusion sampler.

Cumulative weighting and final pruning. We accumulate evidence across scoring steps via an additive criterion $c_i^{(t)} \;=\; c_i^{(t-1)} + \pi_i^{(t)}$, in which $c_i^{(0)} = 0$. At the final scheduled step $t' = \max(\mathcal{S})$, if multiple candidates remain, we prune by selecting the seed with the highest cumulative weight:

$$i^\star \;=\; \arg\max_{i \in \mathcal{I}^{(t')}} \; c_i^{(t')}, \qquad \boldsymbol{z}_0 \;=\; \boldsymbol{z}_0^{(i^\star)}. \tag{18}$$

The surviving latent $\boldsymbol{z}_0$ is then decoded into the final video.

Overall, LatSearch follows the spirit of importance sampling: multiple candidate trajectories are proposed in latent space, weighted according to reward scores that act as surrogate importance factors, and resampled to balance exploration and exploitation. The final pruning step selects the most consistently high-reward trajectory, yielding the decoded video.

4 Experiments

In this section, we evaluate the efficacy of our LatSearch through extensive experiments on a large-scale text-to-video generation task. We will first detail

the implementations and then compare our method to other state-of-the-art inference-time scaling methods for video generation. We finally present the ablation studies. Additional qualitative comparisons and detailed experimental analyses are provided in the *Appendix*.

4.1 Experimental Settings

Implementations. We adopt Qwen2-VL-3B [53] as the backbone for our reward model, owing to its strong performance on multimodal understanding tasks and its suitability for video–language alignment. To construct the latent reward pairs dataset, we first sample 1,000 text prompts that are non-overlapping with VBench-2.0. Using these prompts, we generate 5,000 videos with different random seeds, while also storing the corresponding latents at selected timesteps and their similarity to the final denoised latent. The dataset is partitioned into 80% for training and 20% for testing. For latent reward model training, we initialise the learning rate at 1e-4 and reduce it to 1e-5 at the 10th epoch. The model is trained with a batch size of 4 and stopped at the 15th epoch. Both the regression loss and preference loss are weighted equally with a coefficient of 1.0. We adopt Wan2.1-1.3B [51] as the baseline video generative model, where inference steps are set to 50 and the CFG scale to 5.0 as suggested in [51]. Following [20], each generated video consists of 33 frames at a resolution of 832×480. For LatSearch, the scoring schedule is applied at timesteps 10, 15, and 20. All experiments are conducted on NVIDIA A100 GPUs.

Evaluation. To evaluate the performance of each method, we use VBench-2.0 [62], a comprehensive benchmark that automatically evaluates video generative models for their intrinsic faithfulness. Specifically, VBench-2.0 assesses video performance across five emerging dimensions beyond superficial faithfulness: Human Fidelity, Controllability, Creativity, Physics, and Commonsense. The scores range from 0 to 100, with a higher score indicating better performance in the corresponding aspects. For each noise prior, we generate 3,860 videos for VBench-2.0 evaluation. Detailed implementation settings and evaluation protocols are provided in *Appendix Sec. B*.

4.2 Comparisons to State of the Art

To validate the effectiveness of our method for video inference-time scaling, we first compare it against existing inference-time scaling approaches on VBench-2.0. Specifically, we evaluate against methods that optimise the initial noise, including FreeInit [54] and FreqPrior [59], as well as search-based approaches, including VideoReward [31] and EvoSearch [20]. We report results across the five evaluation dimensions of VBench-2.0, along with their averaged scores, and also provide the inference time of each method. Full per-dimension evaluation results are reported in *Appendix Sec. C.4*.

The main comparison results are reported in Table 1. For methods that optimise the initial noise, although they do not significantly increase computational cost, the absence of an effective verifier limits their effectiveness: simply extracting temporal features from the latent space and feeding them back into the initial noise fails to improve video generation quality. In contrast, search-based

Table 1: Comparison of inference-time scaling methods for video generation on VBench-2.0. The table includes both optimisation-based and search-based approaches. † indicates the use of DPM-Solver++ [32]. **Bold** numbers indicate the best results, while underlined numbers indicate the second-best. ↑ denotes performance degradation and ↓ denotes performance increase.

Methods	Creativity	Commonsense	Controllability	Human Fidelity	Physics	Average	Inference Time (s)
Baseline	53.81	55.63	21.99	82.11	45.98	51.90	77.21 ± 0.26
FreeInit [ECCV'24]	46.80	59.41	22.03	85.22	35.65	$49.82_{(-2.08)}$ ↓	$308.87 \pm 0.22_{(\times 4.00)}$
FreqPrior [ICLR'25]	53.70	55.61	20.35	83.61	38.60	$50.37_{(-1.53)}$ ↓	$142.46 \pm 0.81_{(\times \mathbf{1.85})}$
VideoReward [NeurIPS'25]	55.07	57.91	23.15	82.21	45.67	$52.80_{(+0.90)}$ ↑	$283.63 \pm 2.42_{(\times 3.67)}$
EvoSearch† [arXiv'25]	**59.25**	**61.09**	**26.18**	**86.73**	41.80	$55.01_{(+3.11)}$ ↑	$783.76 \pm 3.15_{(\times 10.15)}$
LatSearch (Ours)	58.12	59.37	22.69	82.59	46.44	$53.84_{(+1.94)}$ ↑	$182.43 \pm 6.53_{(\times \mathbf{2.36})}$
LatSearch† (Ours)	57.53	57.36	23.96	84.01	**53.41**	$\mathbf{55.25}_{(+\mathbf{3.35})}$ ↑	$164.41 \pm 4.79_{(\times \mathbf{2.13})}$

Table 2: Comparison of varied search budgets N on VBench-2.0.

Search Budget N	Creativity	Commonsense	Controllability	Human Fidelity	Physics	Average	Inference Time (s)
Baseline	53.81	55.63	21.99	82.11	45.98	51.90	77.21 ± 0.26
4	57.70	54.70	22.00	83.63	46.03	$52.81_{(+0.91)}$	$132.71 \pm 2.55_{(\times 1.72)}$
6	58.12	**59.37**	**22.69**	82.59	46.44	$53.84_{(+1.94)}$	$182.43 \pm 6.53_{(\times 2.36)}$
8	**58.47**	58.87	22.26	**84.00**	**47.05**	$\mathbf{54.13}_{(+2.23)}$	$225.56 \pm 15.57_{(\times 2.92)}$

approaches rely on output reward optimisation, either via greedy or evolutionary algorithms. While these methods improve quality, they require several times more search time than the baseline, undermining efficiency. Our approach differs in that it evaluates the latent states directly at intermediate steps and employs a probabilistic sampling strategy to retain promising candidate seeds adaptively. As shown in Table 1, our best configuration improves video quality by 3.35% over the baseline, while requiring only 2.13× more computation. Compared to FreqPrior, under the same computational budget, our method achieves a 2.44% higher quality score. When compared with EvoSearch, although our method achieves only a modest 0.24% gain in quality, it is 4.77× faster. Additional qualitative comparisons are provided in *Appendix Sec. D*.

4.3 Ablation Analysis

We conduct ablation studies to better understand the key design choices of LatSearch. Additional analyses on credit assignment strategies, preference loss, temperature sensitivity, and scoring schedules are provided in *Appendix Sec. C.1 and C.2*, together with a human preference study in *Appendix Sec. C.3*.

Effect of Search Budget N. We evaluate how the number of candidate trajectories N influences the performance of LatSearch. As reported in Table 2, increasing the search budget leads to consistent improvements across all VBench-2.0 dimensions. Moving from $N = 4$ to $N = 6$ yields noticeable gains, particularly in Creativity, Commonsense, and Human Fidelity, demonstrating that maintaining a larger pool of latent trajectories enables the search mechanism to explore richer solution paths. Further increasing the budget to $N = 8$ continues to improve performance, indicating that the proposed latent-level scoring and resampling procedure can effectively utilise additional candidates when available. However, the benefits gradually saturate with growing N. Although $N = 8$ produces the strongest results, its marginal improvement over $N = 6$ is smaller compared to the earlier jump from $N = 4$. Importantly, the increase in computational cost remains moderate because evaluations are performed directly in latent space,

Fig. 3: Qualitative comparison with search-based video generation methods. VideoReward achieves strong semantic alignment but suffers from poor temporal dynamics. EvoSearch improves both semantics and dynamics, yet requires heavy search cost. Our LatSearch reaches comparable quality to EvoSearch while being nearly 5× faster. Results are better viewed with zoom-in.

Table 3: Video generation results across different backbones.

Methods	Creativity	Commonsense	Controllability	Human Fidelity	Physics	Average
Wan2.1-14B	55.21	56.18	21.79	89.74	39.96	52.58
+ LatSearch	**56.28**	**57.64**	**22.52**	**91.45**	**40.17**	**53.61**$_{(+1.03)}$
CogVideoX-2B	40.97	37.48	**26.92**	**81.56**	38.74	45.13
+ LatSearch	**49.59**	**45.08**	25.94	77.11	**42.94**	**48.14**$_{(+3.01)}$
CogVideoX-5B	42.52	45.64	26.53	82.22	**43.74**	48.13
+ LatSearch	**50.91**	**48.03**	**27.29**	**83.08**	41.72	**50.21**$_{(+2.08)}$

Table 4: Comparison of VBench-2.0 results under different search strategies and latent reward model settings. A beam-search–style strategy is used for the variant without our PGRP. PL denotes a latent reward model trained with preference loss.

Methods	PL	RGRP	Creativity	Commonsense	Controllability	Human Fidelity	Physics	Average	Inference Time (s)
Baseline	×	×	53.81	55.63	21.99	82.11	45.98	51.90	77.21 (±0.26)
LatSearch	✓	×	56.63$_{(+2.82)}$	57.07$_{(+1.44)}$	22.34$_{(+0.35)}$	82.54$_{(+0.43)}$	43.16$_{(-2.82)}$	52.35$_{(+0.45)}$	171.23(±1.42)$_{(\times 2.22)}$
	×	✓	56.44$_{(+2.64)}$	58.51$_{(+2.89)}$	22.28$_{(+0.29)}$	**84.33**$_{(+2.22)}$	45.54$_{(-0.45)}$	53.42$_{(+1.52)}$	182.43(±6.53)$_{(\times 2.36)}$
	✓	✓	**58.12**$_{(+4.31)}$	**59.37**$_{(+3.74)}$	**22.69**$_{(+0.70)}$	82.59$_{(+0.48)}$	**46.44**$_{(+0.46)}$	**53.84**$_{(+1.94)}$	182.43(±6.53)$_{(\times 2.36)}$

avoiding repeated decoding into video space. This property allows LatSearch to scale gracefully with search budget while remaining computationally practical.

Generalisation Across Backbones. To assess the generality of our approach, we evaluate LatSearch across different video diffusion backbones. Since LatSearch operates in latent space and only requires reward evaluation of intermediate denoising states, it does not depend on architecture-specific components. We first apply LatSearch to the larger Wan2.1-14B backbone. As shown in Table 3, LatSearch improves the average VBench-2.0 score from 52.58 to 53.61, with gains across all five evaluation dimensions. This indicates that latent reward guidance remains effective when scaling to a stronger generation backbone.

To further evaluate whether LatSearch generalises beyond the Wan model family, we apply it to two CogVideoX backbones, CogVideoX-2B and CogVideoX-5B. As shown in Table 3, LatSearch improves the average score from 45.13 to 48.14 on CogVideoX-2B, yielding a gain of +3.01, and from 48.13 to 50.21 on CogVideoX-5B, yielding a gain of +2.08. Although the improvements vary across individual dimensions, the consistent average gains across both CogVideoX models demonstrate that LatSearch is not tied to a specific video diffusion architecture and can provide effective scaling across different model families.

Effectiveness of the RGRP. We compare RGRP with a beam search–style approach that relies solely on cumulative reward scores. Concretely, both methods follow the same setting of scoring latent candidates at scheduled denoising

Table 5: VBench-2.0 evaluation results under different credit assignment strategies.

Methods	Creativity	Commonsense	Controllability	Human Fidelity	Physics	Average
Baseline	53.81	55.63	21.99	82.11	45.98	51.90
Uniform	54.98	52.40	21.44	84.98	47.75	52.31$_{(+0.41)}$
Exponential	54.50	52.09	21.63	83.03	49.30	52.11$_{(+0.21)}$
L2 Error	55.78	56.64	22.20	82.12	46.95	52.74$_{(+0.84)}$
Cosine Similarity	58.12	59.37	22.69	82.59	46.44	**53.84**$_{(+1.94)}$

Table 6: Results of LatSearch across different video lengths and resolutions.

Setting	Methods	Creativity	Commonsense	Controllability	Human Fidelity	Physics	Average
2s (32-frame)	Wan-1.3B	53.81	55.63	21.99	82.11	45.98	51.90
	+ LatSearch	**58.12**	**59.37**	**22.69**	**82.59**	**46.44**	**53.84**$_{(+1.94)}$
3s (48-frame)	Wan-1.3B	54.41	**54.10**	21.11	81.72	41.97	50.66
	+ LatSearch	**55.00**	53.23	**21.70**	**82.18**	**47.23**	**51.87**$_{(+1.21)}$
4s (64-frame)	Wan-1.3B	**53.35**	50.96	22.47	79.03	**45.95**	50.35
	+ LatSearch	52.87	**56.16**	**23.18**	**80.23**	43.66	**51.22**$_{(+0.87)}$
5s (80-frame)	Wan-1.3B	**54.88**	53.25	**23.60**	77.38	39.88	49.80
	+ LatSearch	54.01	**55.87**	22.59	**80.74**	**42.65**	**51.17**$_{(+1.37)}$
480p (832×480)	Wan-1.3B	53.81	55.63	21.99	82.11	45.98	51.90
	+ LatSearch	**58.12**	**59.37**	**22.69**	**82.59**	**46.44**	**53.84 (+1.94)**
720p (1280×720)	Wan-1.3B	41.52	**59.07**	**22.29**	**81.44**	37.67	48.40
	+ LatSearch	**42.12**	57.63	22.09	81.12	**43.03**	**49.20 (+0.80)**

timesteps. The difference is that the beam search baseline retains a fixed number of seeds purely based on accumulated rewards, while our RGRP method incorporates probabilistic resampling and pruning guided by reward signals. As shown in Table 4, RGRP achieves consistently better results across most of the five evaluation dimensions. On average, it improves video quality by 0.42% compared to the beam search–style baseline, while incurring only a marginal increase in computation. These results highlight the benefit of probabilistic selection in balancing exploration and exploitation, leading to more robust search outcomes than deterministic reward accumulation. This demonstrates that RGRP mitigates overfitting to accumulated reward signals and promotes more diverse yet high-quality candidate selection.

Different Credit Assignment Strategies. To investigate how reward signals should be propagated across denoising steps, we evaluate several credit assignment strategies, including Uniform, Exponential, L2 Error, and Cosine Similarity weighting. As shown in Table 5, Cosine Similarity achieves the best overall performance, improving the VBench-2.0 average score from 51.90 to 53.84. This strategy notably enhances Creativity and Commonsense while maintaining stable performance across Controllability and Physical plausibility. In contrast, Uniform and Exponential weighting provide only marginal gains, suggesting that naïvely distributing rewards across timesteps is insufficient. We hypothesise that Cosine Similarity better aligns latent updates with semantic consistency, enabling more effective guidance during latent reward model training. Additional empirical approximation-error analysis is provided in *Appendix Sec. C.1.*

Scalability to Video Lengths and Resolutions. We also examine LatSearch under different video lengths and resolutions. Table 6 reports results

for 2s/32-frame, 3s/48-frame, 4s/64-frame, and 5s/80-frame generation, as well as 480p and 720p resolutions. LatSearch improves the average VBench-2.0 score by +1.94, +1.21, +0.87, and +1.37 for 2s, 3s, 4s, and 5s videos, respectively. It also improves performance at both 480p and 720p, with gains of +1.94 and +0.80. These results suggest that LatSearch remains effective beyond the default evaluation setting, although the gains become smaller for more challenging longer-duration and higher-resolution generation.

Effectiveness of the Preference Loss. To validate the effectiveness of incorporating preference loss into the latent reward model, we conduct experiments from two perspectives: (1) the consistency between the latent reward and the video-level verifier, and (2) the improvement in video generation quality under our proposed LatSearch framework. For the first perspective, we construct approximately 446K latent pairs in the test set, with preferences computed from the corresponding video scores and similarity to the final clean latent. As shown in Figure 4, compared with regression loss, preference loss improves the alignment accuracy by 3.1%, 2.65%, 2.96%, 3.54%, and 3.21% at denoising steps of 10, 15, 20, 25, and 30, respectively. This demonstrates the effectiveness of preference learning. Furthermore, we directly validate the preference loss within our latent search method without RGRP. As shown in Table 4, the use of preference loss leads to a 1.07% improvement in video quality. This further highlights the effectiveness of our search algorithm: as the latent reward model becomes stronger, our method consistently achieves better performance. These results confirm that preference-based supervision is not only beneficial at the model level but also translates into measurable improvements in downstream video generation.

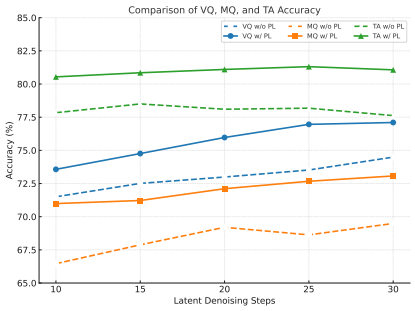

Fig. 4: Comparison of VQ, MQ, and TA accuracy across different loss function settings and denoising steps.

Inference Time Comparison. Table 7 presents a detailed runtime breakdown across different methods. VideoReward and EvoSearch incur substantial computational overhead due to repeated full video decoding and the maintenance of multiple candidate trajectories, resulting in significantly increased DiT and decoder costs. For example, EvoSearch requires 756.66 seconds of DiT computation and 31.99 seconds of decoding on average, leading to a total runtime of 790.57 seconds.

In contrast, LatSearch operates directly in the latent space, keeping decoder usage nearly identical to the baseline (1.88 vs. 1.85 seconds) and introducing only a lightweight latent reward evaluation cost with 1.03 seconds. This efficiency stems from the low overhead of latent-space operations: each DiT forward step takes only 1.35 seconds, while latent reward evaluation requires 0.84 seconds per latent, substantially cheaper than full video decoding. As a result, LatSearch achieves the best VBench-2.0 performance at 55.25% while requiring

Table 7: Inference time comparison across different methods.

Methods	DiT Time (↓)	Decoder Time (↓)	Reward Time (↓)	Total Time (↓)	VBench-2.0 Results (↑)
Baseline	67.46 ± 0.71	1.85 ± 0.26	0	69.31 ± 0.76	51.90
VideoReward	266.71 ± 0.40	10.12 ± 0.54	0.56 ± 0.33	277.39 ± 0.75	52.80
EvoSearch	756.66 ± 1.27	31.99 ± 0.96	1.92 ± 1.04	790.57 ± 1.90	55.01
LatSearch	156.11 ± 5.11	1.88 ± 0.24	1.03 ± 0.06	159.02 ± 5.12	**55.25**

only 159.02 seconds in total runtime, which is nearly 5× faster than EvoSearch and substantially more efficient than VideoReward, requiring 277.39 seconds.

5 Conclusion

We have presented LatSearch, a new inference-time scaling framework for video diffusion that addresses the limitations of existing methods, which either impose priors on initial noise or evaluate only final decoded videos. By introducing a latent reward model for intermediate evaluation and integrating it with Reward-Guided Resampling and Pruning (RGRP), LatSearch achieves both higher video quality and improved efficiency. Specifically, it attains 2.44% higher fidelity quality under equal compute cost, and is 4.77× faster when achieving the same fidelity quality, compared to state-of-the-art methods. Experiments on VBench-2.0 demonstrate consistent improvements across diverse evaluation dimensions, highlighting latent reward guidance as a promising direction for scalable and efficient video generation.

6 Limitations & Future Work

Limitations. While LatSearch demonstrates consistent performance improvements and considerable inference-time savings, several limitations remain. Firstly, our resampling-and-pruning procedure is inspired by Sequential Monte Carlo methods, yet theoretical convergence guarantees are difficult to establish due to the learned and approximate nature of the latent reward model. Consequently, although empirically effective, we do not claim formal optimality of the search procedure. Secondly, we rely on cosine-similarity weighting to transform video-level rewards into latent-level supervision. Although ablations show this strategy performs best among tested alternatives, it remains an approximation of true semantic contribution. More principled or learned temporal credit-assignment functions may further improve latent-reward consistency.

Future Work. Firstly, developing a lightweight temporal similarity estimator, potentially contrastive or self-supervised, could provide a more accurate credit-assignment mechanism, directly addressing the current approximation bottleneck. Secondly, since two dimensions of our reward model (visual quality and motion quality) are modality-agnostic, LatSearch can be extended to audio–video generation, video editing, and instruction-guided transformations, by replacing the text-alignment objective with a task-specific alignment module.

Acknowledgments. This research utilised Queen Mary's Apocrita HPC facility, supported by QMUL Research-IT. Zengqun Zhao was funded by Queen Mary Principal's PhD Studentships. This work was supported by Huawei Technologies Germany (Project 19655832: Uncertainty Aware Data Attribution Of Vision-Language Models).

References

1. Agarwal, N., Ali, A., Bala, M., Balaji, Y., Barker, E., Cai, T., Chattopadhyay, P., Chen, Y., Cui, Y., Ding, Y., et al.: Cosmos world foundation model platform for physical ai. arXiv preprint arXiv:2501.03575 (2025) 2
2. Ban, Y., Wang, R., Zhou, T., Gong, B., Hsieh, C.J., Cheng, M.: The crystal ball hypothesis in diffusion models: Anticipating object positions from initial noise. In: ICLR (2025) 2, 4
3. Bao, F., Nie, S., Xue, K., Cao, Y., Li, C., Su, H., Zhu, J.: All are worth words: A vit backbone for diffusion models. In: CVPR. pp. 22669–22679 (2023) 4
4. Bhagat, S., Uppal, S., Yin, Z., Lim, N.: Disentangling multiple features in video sequences using gaussian processes in variational autoencoders. In: ECCV. pp. 102–117. Springer (2020) 4
5. Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al.: Stable video diffusion: Scaling latent video diffusion models to large datasets. arXiv preprint arXiv:2311.15127 (2023) 4
6. Brooks, T., Hellsten, J., Aittala, M., Wang, T.C., Aila, T., Lehtinen, J., Liu, M.Y., Efros, A., Karras, T.: Generating long videos of dynamic scenes. NeurIPS **35**, 31769–31781 (2022) 4
7. Burgert, R., Xu, Y., Xian, W., Pilarski, O., Clausen, P., He, M., Ma, L., Deng, Y., Li, L., Mousavi, M., et al.: Go-with-the-flow: Motion-controllable video diffusion models using real-time warped noise. In: CVPR. pp. 13–23 (2025) 2, 5
8. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: ICCV. pp. 9650–9660 (2021) 5
9. Chang, P., Tang, J., Gross, M., Azevedo, V.C.: How i warped your noise: a temporally-correlated noise prior for diffusion models. In: ICLR (2024) 2, 5
10. Chen, H., Xia, M., He, Y., Zhang, Y., Cun, X., Yang, S., Xing, J., Liu, Y., Chen, Q., Wang, X., et al.: Videocrafter1: Open diffusion models for high-quality video generation. arXiv preprint arXiv:2310.19512 (2023) 4
11. Chen, S., Xu, M., Ren, J., Cong, Y., He, S., Xie, Y., Sinha, A., Luo, P., Xiang, T., Perez-Rua, J.M.: Gentron: Diffusion transformers for image and video generation. In: CVPR. pp. 6441–6451 (2024) 4
12. Chen, W., Ji, Y., Wu, J., Wu, H., Xie, P., Li, J., Xia, X., Xiao, X., Lin, L.: Control-a-video: Controllable text-to-video diffusion models with motion prior and reward feedback learning. arXiv preprint arXiv:2305.13840 (2023) 4
13. Dalal, K., Koceja, D., Xu, J., Zhao, Y., Han, S., Cheung, K.C., Kautz, J., Choi, Y., Sun, Y., Wang, X.: One-minute video generation with test-time training. In: CVPR. pp. 17702–17711 (2025) 2, 4
14. Deng, H., Pan, T., Diao, H., Luo, Z., Cui, Y., Lu, H., Shan, S., Qi, Y., Wang, X.: Autoregressive video generation without vector quantization. arXiv preprint arXiv:2412.14169 (2024) 4
15. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. NeurIPS **34**, 8780–8794 (2021) 6
16. Ding, Z., Jin, C., Liu, D., Zheng, H., Singh, K.K., Zhang, Q., Kang, Y., Lin, Z., Liu, Y.: Dollar: Few-step video generation via distillation and latent reward optimization. arXiv preprint arXiv:2412.15689 (2024) 4
17. Doucet, A., De Freitas, N., Gordon, N.: An introduction to sequential monte carlo methods. In: Sequential Monte Carlo methods in practice, pp. 3–14. Springer (2001) 9

18. Gu, Y., Mao, W., Shou, M.Z.: Long-context autoregressive video modeling with next-frame prediction. arXiv preprint arXiv:2503.19325 (2025) 4
19. Guo, Y., Yang, C., Rao, A., Liang, Z., Wang, Y., Qiao, Y., Agrawala, M., Lin, D., Dai, B.: Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. In: ICLR (2024) 4
20. He, H., Liang, J., Wang, X., Wan, P., Zhang, D., Gai, K., Pan, L.: Scaling image and video generation via test-time evolutionary search. arXiv preprint arXiv:2505.17618 (2025) 2, 3, 5, 10
21. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. NeurIPS **30** (2017) 5
22. Hiranaka, A., Chen, S.F., Lai, C.H., Kim, D., Murata, N., Shibuya, T., Liao, W.H., Sun, S.H., Mitsufuji, Y.: Hero: Human-feedback efficient reinforcement learning for online diffusion model finetuning. In: ICLR (2024) 4
23. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022) 6
24. Hsieh, J.T., Liu, B., Huang, D.A., Fei-Fei, L.F., Niebles, J.C.: Learning to decompose and disentangle representations for video prediction. NeurIPS **31** (2018) 4
25. Kara, O., Kurtkaya, B., Yesiltepe, H., Rehg, J.M., Yanardag, P.: Rave: Randomized noise shuffling for fast and consistent video editing with diffusion models. In: CVPR. pp. 6507–6516 (2024) 2
26. Karras, J., Holynski, A., Wang, T.C., Kemelmacher-Shlizerman, I.: Dreampose: Fashion image-to-video synthesis via stable diffusion. In: ICCV. pp. 22623–22633 (2023) 2
27. Kim, K., Kim, S.: Model already knows the best noise: Bayesian active noise selection via attention in video diffusion model. arXiv preprint arXiv:2505.17561 (2025) 2, 3, 4, 5
28. Kong, W., Tian, Q., Zhang, Z., Min, R., Dai, Z., Zhou, J., Xiong, J., Li, X., Wu, B., Zhang, J., et al.: Hunyuanvideo: A systematic framework for large video generative models. arXiv preprint arXiv:2412.03603 (2024) 4
29. Liao, X., Wei, W., Qu, X., Cheng, Y.: Step-level reward for free in rl-based t2i diffusion model fine-tuning. arXiv preprint arXiv:2505.19196 (2025) 3
30. Liu, F., Wang, H., Cai, Y., Zhang, K., Zhan, X., Duan, Y.: Video-t1: Test-time scaling for video generation. arXiv preprint arXiv:2503.18942 (2025) 2, 4
31. Liu, J., Liu, G., Liang, J., Yuan, Z., Liu, X., Zheng, M., Wu, X., Wang, Q., Qin, W., Xia, M., et al.: Improving video generation with human feedback. In: NeurIPS (2025) 3, 4, 5, 7, 10
32. Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., Zhu, J.: Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. Machine Intelligence Research pp. 1–22 (2025) 11
33. Luo, Y., Zhao, X., Chen, M., Zhang, K., Shao, W., Wang, K., Wang, Z., You, Y.: Enhance-a-video: Better generated video for free. arXiv preprint arXiv:2502.07508 (2025) 4
34. Ma, N., Tong, S., Jia, H., Hu, H., Su, Y.C., Zhang, M., Yang, X., Li, Y., Jaakkola, T., Jia, X., et al.: Scaling inference time compute for diffusion models. In: CVPR. pp. 2523–2534 (2025) 2, 3, 4, 5
35. Ma, Y., Chen, J., Di, D., Xie, Q., Fan, L., Chen, W., Gou, X., Zhao, N., Yang, X.: Tuning-free long video generation via global-local collaborative diffusion. arXiv preprint arXiv:2501.05484 (2025) 4

36. Nam, H., Kim, J., Lee, D., Ye, J.C.: Optical-flow guided prompt optimization for coherent video generation. In: CVPR. pp. 7837–7846 (2025) 4
37. Oshima, Y., Suzuki, M., Matsuo, Y., Furuta, H.: Inference-time text-to-video alignment with diffusion latent beam search. In: NeurIPS (2025) 2, 3, 5
38. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. NeurIPS **35**, 27730–27744 (2022) 8
39. Ouyang, W., Xiao, Z., Yang, D., Zhou, Y., Yang, S., Yang, L., Si, J., Pan, X.: Tokensgen: Harnessing condensed tokens for long video generation. arXiv preprint arXiv:2507.15728 (2025) 4
40. Peebles, W., Xie, S.: Scalable diffusion models with transformers. In: ICCV. pp. 4195–4205 (2023) 4
41. Peruzzo, E., Xu, D., Xu, X., Shi, H., Sebe, N.: Ragme: Retrieval augmented video generation for enhanced motion realism. In: ICMR. pp. 1081–1090 (2025) 4
42. Qi, Z., Bai, L., Xiong, H., Xie, Z.: Not all noises are created equally: Diffusion noise selection and optimization. arXiv preprint arXiv:2407.14041 (2024) 2, 4
43. Qiu, H., Xia, M., Zhang, Y., He, Y., Wang, X., Shan, Y., Liu, Z.: Freenoise: Tuning-free longer video diffusion via noise rescheduling. In: ICLR (2023) 4
44. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML. pp. 8748–8763 (2021) 5
45. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. NeurIPS **29** (2016) 5
46. Singhal, R., Horvitz, Z., Teehan, R., Ren, M., Yu, Z., McKeown, K., Ranganath, R.: A general framework for inference-time scaling and steering of diffusion models. arXiv preprint arXiv:2501.06848 (2025) 3, 5
47. Skorokhodov, I., Tulyakov, S., Elhoseiny, M.: Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In: CVPR. pp. 3626–3636 (2022) 4
48. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020) 8
49. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. In: ICLR (2020) 6
50. Sun, Y., Wu, J., Cao, Y., Xu, C., Wang, Y., Cao, W., Luo, D., Wang, C., Fu, Y.: Swiftvideo: A unified framework for few-step video generation through trajectory-distribution alignment. arXiv preprint arXiv:2508.06082 (2025) 4
51. Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., et al.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025) 4, 10
52. Wang, J., Yuan, H., Chen, D., Zhang, Y., Wang, X., Zhang, S.: Modelscope text-to-video technical report. arXiv preprint arXiv:2308.06571 (2023) 4
53. Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., et al.: Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. arXiv preprint arXiv:2409.12191 (2024) 10
54. Wu, T., Si, C., Jiang, Y., Huang, Z., Liu, Z.: Freeinit: Bridging initialization gap in video diffusion models. In: ECCV. pp. 378–394 (2024) 2, 5, 10
55. Xiao, Z., Zhou, Y., Yang, S., Pan, X.: Video diffusion models are training-free motion interpreter and controller. NeurIPS **37**, 76115–76138 (2024) 4

56. Yang, H., Tang, F., Hu, M., Yin, Q., Li, Y., Liu, Y., Peng, Z., Gao, P., He, J., Ge, Z., et al.: Scalingnoise: Scaling inference-time search for generating infinite videos. arXiv preprint arXiv:2503.16400 (2025) 2, 3, 5
57. Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., Xu, J., Yang, Y., Hong, W., Zhang, X., Feng, G., et al.: Cogvideox: Text-to-video diffusion models with an expert transformer. In: ICLR (2024) 4
58. Yuan, H., Zhang, S., Wang, X., Wei, Y., Feng, T., Pan, Y., Zhang, Y., Liu, Z., Albanie, S., Ni, D.: Instructvideo: Instructing video diffusion models with human feedback. In: CVPR. pp. 6463–6474 (2024) 4
59. Yuan, Y., Guo, Y., Wang, C., Zhang, W., Xu, H., Zhang, L.: Freqprior: Improving video diffusion models with frequency filtering gaussian noise. In: ICLR (2025) 2, 5, 10
60. Zhang, X., Duan, Z., Gong, D., Liu, L.: Training-free motion-guided video generation with enhanced temporal consistency using motion consistency loss. arXiv preprint arXiv:2501.07563 (2025) 2, 5
61. Zhao, W., Bai, L., Rao, Y., Zhou, J., Lu, J.: Unipc: A unified predictor-corrector framework for fast sampling of diffusion models. NeurIPS **36**, 49842–49869 (2023) 6, 8
62. Zheng, D., Huang, Z., Liu, H., Zou, K., He, Y., Zhang, F., Zhang, Y., He, J., Zheng, W.S., Qiao, Y., et al.: Vbench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness. arXiv preprint arXiv:2503.21755 (2025) 10
63. Zhou, Z., Shao, S., Bai, L., Zhang, S., Xu, Z., Han, B., Xie, Z.: Golden noise for diffusion models: A learning framework. arXiv preprint arXiv:2411.09502 (2024) 2, 4
64. Zhu, B., Jiang, Y., Xu, B., Yang, S., Yin, M., Wu, Y., Sun, H., Wu, Z.: Aligning anime video generation with human feedback. arXiv preprint arXiv:2504.10044 (2025) 4