STARFlow-V: End-to-End Video Generative Modeling with Normalizing Flows

Jiatao Gu, **Ying Shen**, **Tianrong Chen**, **Laurent Dinh**, **Yuyang Wang**, **Miguel Ángel Bautista**, **David Berthelot**, **Josh Susskind**, **Shuangfei Zhai**

Apple

Normalizing flows (NFs) are end-to-end likelihood-based generative models for continuous data, and have recently regained attention with encouraging progress on image generation. Yet in the video generation domain, where spatiotemporal complexity and computational cost are substantially higher, state-of-the-art systems almost exclusively rely on diffusion-based models. In this work, we revisit this design space by presenting STARFlow-V, a normalizing flow-based video generator with substantial benefits such as end-to-end learning, robust causal prediction, and native likelihood estimation. Building upon the recently proposed STARFlow, STARFlow-V operates in the spatiotemporal latent space with a global-local architecture which restricts causal dependencies to a global latent space while preserving rich local within-frame interactions. This eases error accumulation over time, a common pitfall of standard autoregressive diffusion model generation. Additionally, we propose *flow-score matching*, which equips the model with a light-weight causal denoiser to improve the video generation consistency in an autoregressive fashion. To improve the sampling efficiency, STARFlow-V employs a video-aware Jacobi iteration scheme that recasts inner updates as parallelizable iterations without breaking causality. Thanks to the invertible structure, the same model can natively support text-to-video, image-to-video as well as video-to-video generation tasks. Empirically, STARFlow-V achieves strong visual fidelity and temporal consistency with practical sampling throughput relative to diffusion-based baselines. These results present the first evidence, to our knowledge, that NFs are capable of high-quality autoregressive video generation, establishing them as a promising research direction for building world models.

Code: https://github.com/apple/ml-starflow
Correspondence: Jiatao Gu (jgu32@apple.com)
Date: November 25, 2025

1 Introduction

Deep generative modeling has advanced rapidly with breakthroughs across language (Achiam et al., 2023; OpenAI, 2024a), images (Podell et al., 2023; Batifol et al., 2025; Wu et al., 2025), and videos (OpenAI, 2024b; Wan et al., 2025; DeepMind, 2025) domains. Among these modalities, *video generation* is uniquely demanding: beyond high perceptual quality, models must capture rich spatiotemporal structure, remain robust over long horizons, and often operate under causal constraints for streaming and interactive use. Such capabilities are central not only to creative media (Ye et al., 2025; Yuan et al., 2025), but also to emerging *world models* for gaming, simulation and embodied AI (Ha and Schmidhuber, 2018; Yang et al., 2023; Hu et al., 2023; Google DeepMind, 2024; Hafner et al., 2025).

Recent scaling of data, model capacity, and compute has pushed video generation to new levels of fidelity (Yang et al., 2025; Kong et al., 2024; Kondratyuk et al., 2024; Yu et al., 2024; Wan et al., 2025; Seawead et al., 2025; Gao et al., 2025). In this space, *diffusion-based* approaches (Ho et al., 2020; Rombach et al., 2022; Peebles and Xie, 2023; Lipman et al., 2023; Esser et al., 2024) have emerged as the dominant

Work done while JG holding a joint affiliation with University of Pennsylvania, and YS working as a research intern at Apple MLR.

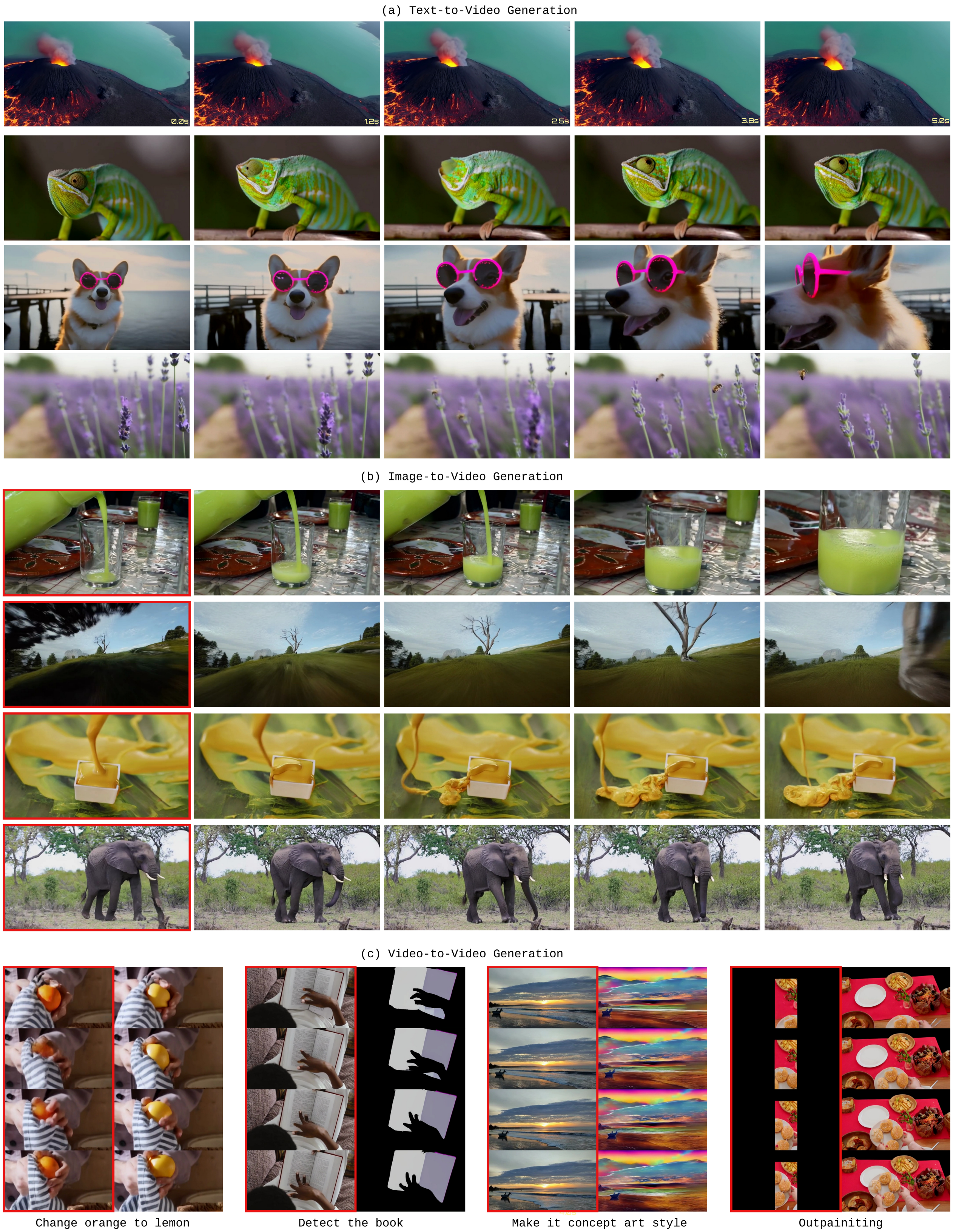


Figure 1 Samples from STARFlow-V across three tasks. All videos are 480 p at 16 fps. Red boxes mark the conditioning inputs. The same autoregressive architecture is used for all tasks with no task-specific modifications. **Please find more generated videos and comparisons in the released code https://github.com/apple/ml-starflow.**

backbone for text- and image-conditioned video synthesis, thanks to their strong empirical performance and flexible conditioning mechanisms. Standard diffusion models are trained by corrupting frames with noise drawn from a schedule and learning a denoiser that inverts this process one step at a time, which leads to an iterative sampling procedure at inference. For offline generation this formulation works well, but the parallel denoising of multiple frames is inherently non-causal: future frames can influence earlier ones, making it less natural to apply in streaming or interactive settings that require strictly causal rollouts. Causally conditioned and sequential diffusion variants (Chen et al., 2024a; Huang et al., 2025) mitigate some of these issues, but still inherit the need to simulate noise at different timesteps and frames during training and can exhibit train–test mismatch during long-horizon autoregressive generation.

In parallel, *normalizing flows* (NFs) (Rezende and Mohamed, 2015; Dinh et al., 2014, 2016) offer a distinct, likelihood-based alternative. NFs are continuous end-to-end generative models that provide exact log-likelihood evaluation, non-iterative sampling, and native support for invertible feature mappings. After an initial wave of work (Dinh et al., 2016; Kingma and Dhariwal, 2018), they received relatively less attention compared to diffusion models, but have recently regained interest with encouraging progress on image generation (Zhai et al.; Gu et al., 2025; Zheng et al., 2025). In particular, STARFlow (Gu et al., 2025) shows that parameterizing an "autoregressive normalizing flow" with a Transformer and operating in a latent space allows flows to scale competitively in the high-resolution image domain. Yet, in the video domain—where complexity and computational cost are substantially higher—state-of-the-art systems almost exclusively rely on diffusion, and it remains unclear whether NFs can be practical for video.

In this work, we revisit this design space and introduce STARFlow-V, a normalizing-flow-based video generator that combines end-to-end training with causal, likelihood-based modeling. Building on STARFlow (Gu et al., 2025), STARFlow-V operates in a spatiotemporal latent space with a *global–local* architecture: a compact global latent sequence carries long-range temporal context, while local latent blocks preserve fine-grained within-frame structure. By delegating temporal reasoning to this high-level space, the model mitigates the accumulation of autoregressive errors that commonly plagues diffusion-based video generators. As observed in TARFlow (Zhai et al., 2024), training flows on slightly perturbed data with a subsequent denoising step can significantly improve robustness. Unlike existing methods (Zhai et al., 2024; Gu et al., 2025), we propose *flow-score matching*, which learns a lightweight causal denoiser to enhance temporal consistency in video scenarios. To further improve efficiency, STARFlow-V employs a video-aware Jacobi-style update scheme that recasts inner refinement steps as parallelizable iterations. Finally, owing to its invertible nature, the same backbone naturally supports text-to-video (T2V), image-to-video (I2V), and video-to-video (V2V) generation by simply changing the form of the conditioning signal.

Across all benchmarks, STARFlow-V attains visually coherent and temporally stable generations while maintaining practical sampling speed relative to diffusion-based models. We believe this provides **initial** evidence that NFs are capable of high-quality autoregressive video generation and potentially world models.

2 Background

2.1 Video Generative Models

Given N frames $\boldsymbol{x}_{1:N} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ and optional conditioning C (*e.g.*, text, image, audio, layout, camera), video generative models seek to model the joint distribution of all frames $p(\boldsymbol{x}_{1:N} \mid C)$ and sample novel videos from the learned model. While earlier work explored GANs (Vondrick et al., 2016; Tulyakov et al., 2018; Skorokhodov et al., 2022), VAEs (Babaeizadeh et al., 2018; Castrejon et al., 2019; Wu et al., 2021), and discrete autoregressive models (Yan et al., 2021; Yu et al., 2024; Kondratyuk et al., 2024), the field has largely converged on diffusion-based methods Ho et al. (2022c,a). Spurred by the release of Sora (Brooks et al., 2024), DiT-style approaches (Peebles and Xie, 2023) have shown strong generalization at scale (Gao et al., 2025; Wan et al., 2025; DeepMind, 2025). A key distinction from prior paradigms is that training of diffusion-based models is *Not End-to-End*: diffusion-based models corrupt frames with noise at randomly sampled levels and train a denoiser to invert this process, optimizing an objective closely related to the lower bound of $\log p(\boldsymbol{x}_{1:N} \mid C)$. This setup incurs high cost—especially for video—as each update supervises only a single noise level. At inference time, one sample is generated by iteratively denoising from Gaussian noise.

Diffusion-based video generation is typically non-causal: all frames are corrupted with noise and denoised in parallel (Ho et al., 2022c). Yet many real-world applications demand causal, often interactive synthesis (*e.g.*, online streaming, video games, robotics), where frames must be produced sequentially. Autoregressive (AR) diffusion models (Chen et al., 2024a; Song et al., 2025; Yin et al., 2025)—a line of work that combines chain-rule factorization with diffusion—aim to alleviate prior limitations by introducing asynchronous, frame-wise noise schedules during training, modeling each conditional $p(\boldsymbol{x}_n \mid \boldsymbol{x}_{<n})$ as a diffusion process. Despite their strengths, AR generation typically suffers from *exposure bias*: during training, models condition on ground-truth contexts, whereas at inference they must rely on their own (imperfect) predictions. This train–test mismatch compounds over time, degrading long-horizon video quality. The *non–end-to-end* nature of diffusion training further exacerbates this gap, though recent efforts such as Self-Forcing (Huang et al., 2025) seek to mitigate it via sequential post-training with distillation objectives. However, they are not readily applicable in the pre-training stage on raw video data.

2.2 Autoregressive Normalizing Flows

Normalizing flows (NFs; Rezende and Mohamed, 2015; Dinh et al., 2014, 2016; Kingma and Dhariwal, 2018; Ho et al., 2019) are likelihood-based generative models built from invertible transformations. Given a continuous input $\boldsymbol{x} \sim p_{\text{data}}$, $\boldsymbol{x} \in \mathbb{R}^D$, an NF learns a bijection $f_\theta : \mathbb{R}^D \to \mathbb{R}^D$ that maps data $\boldsymbol{x}$ to latents $\boldsymbol{z} = f_\theta(\boldsymbol{x})$. Unlike diffusion models, NFs are trained *end-to-end* via a tractable maximum-likelihood objective derived from the change-of-variables formula:

$$\mathcal{L}_{\text{NF}}(\theta) = \mathbb{E}_{\boldsymbol{x}}\left[\log p_0\big(f_\theta(\boldsymbol{x})\big) + \log\left|\det(J_{f_\theta}(\boldsymbol{x}))\right|\right], \tag{2.1}$$

where the first term encourages mapping data to high-density regions of a simple prior p_0 (e.g., standard Gaussian), and the Jacobian term J_f accounts for the local volume change induced by f_θ, preventing collapse. Once trained, sampling is immediate via inversion: draw $\boldsymbol{z} \sim p_0(\boldsymbol{z})$ and set $\boldsymbol{x} = f_\theta^{-1}(\boldsymbol{z})$. Historically, however, NFs have been viewed as less competitive than diffusion models due to architectural rigidity and training instability (Dinh et al., 2016).

Recently, TARFlow (Zhai et al.) and its scalable extension, STARFlow (Gu et al., 2025), have revisited normalizing flows as next-generation backbones for generative modeling. Both methods instantiate autoregressive flows (AFs) (Kingma et al., 2016; Papamakarios et al., 2017)—NFs whose invertible transformations are parameterized autoregressively—and use causal Transformer blocks, in the style of LLMs, as their primary building units. Formally, STARFlow (Gu et al., 2025) stacks T autoregressive flow blocks with alternating directions, where each block applies an affine transform whose parameters are predicted by a causal Transformer under a (self-exclusive) causal mask $\boldsymbol{m}$:

$$\boldsymbol{z} \;=\; \big[\boldsymbol{x} - \mu_\theta(\boldsymbol{x} \odot \boldsymbol{m})\big] / \sigma_\theta(\boldsymbol{x} \odot \boldsymbol{m}), \;\; \sigma_\theta(\cdot) > 0, \tag{2.2}$$

where $\boldsymbol{x}, \boldsymbol{z}$ are the input and output of each block, $\odot$ denotes the Hadamard product. As shown in STARFlow (Gu et al., 2025), $T \geq 3$ blocks suffice for universal density modeling where masks alternate between left-to-right ($\rightarrow$) and right-to-left ($\leftarrow$) to capture bidirectional dependencies.

Despite STARFlow demonstrating competitive quality with state-of-the-art diffusion (Podell et al., 2023; Esser et al., 2024) on large-scale text-to-image tasks, evidence for normalizing flows in video generation remains sparse. To our best knowledge, the only prior NF-based video model is VideoFlow (Kumar et al., 2019), which builds on Glow (Kingma and Dhariwal, 2018) and is constrained by limited capacity, low resolution, and domain-specific settings. Compared to images, video generation is substantially more challenging for NFs due to higher spatiotemporal dimensionality. Nevertheless, we argue that normalizing flows—exemplified by STARFlow—are a natural fit for video modeling, especially in autoregressive settings.

3 STARFlow-V

We propose STARFlow-V, a novel paradigm for video generation based on normalizing flows. While inspired by STARFlow (Gu et al., 2025), STARFlow-V is not a direct port to the video domain; it introduces several architectural redesigns and algorithmic innovations tailored to spatiotemporal data. In what follows, we present the architecture and its autoregressive formulation (Section 3.1), the training procedure (Section 3.2), the inference pipeline (Section 3.3), and applications enabled by our model (Section 3.4).

3.1 Proposed Model

For a video $\boldsymbol{x} \in \mathbb{R}^{N\times H\times W\times D}$, each frame $\boldsymbol{x}_n$ is flattened to $\mathbb{R}^{HW\times D}$, $\boldsymbol{x}_n = (\boldsymbol{x}_{n,1}, \ldots, \boldsymbol{x}_{n,HW})$, and all frames are concatenated into a sequence of NHW tokens. We operate in a compressed latent space using a pretrained 3D causal VAE (Wan et al., 2025). STARFlow-V models the joint distribution $p_\theta(\boldsymbol{x})$ via an invertible mapping f_θ implemented as autoregressive flows (Equation (2.2)). Following Gu et al. (2025), we use a *deep–shallow* decomposition $f_\theta = f_D \circ f_S$, where a small stack of *shallow* flow blocks with alternating (left-to-right / right-to-left) masks maps $\boldsymbol{x}$ to intermediate latents $\boldsymbol{u} = f_S(\boldsymbol{x})$, and a *deep* causal-Transformer flow f_D then maps $\boldsymbol{u}$ to the prior, producing $\boldsymbol{z} = f_D(\boldsymbol{u})$. By the change-of-variables formula,

$$p_\theta(\boldsymbol{x}) \;=\; p_0(\boldsymbol{z})\,\left|\det J_{f_D}(\boldsymbol{u})\right|\,\left|\det J_{f_S}(\boldsymbol{x})\right|, \tag{3.1}$$

where p_0 is a simple prior (e.g., standard Gaussian). Most capacity is allocated to the deep block f_D for semantic modeling, while the shallow stack f_S handles local reshaping. For videos, we can simply treat all frames as one long token sequence: f_D follows a left-to-right causal order over the video (causal across frames, raster order within each frame), and f_S retains the alternating masks defined above. Because f_S propagates information from future frames to past ones, this naïve design yields a *non-causal* video generator, motivating the global–local restructuring described next.

Global–Local Architecture Observing that f_D is inherently autoregressive and that f_S mainly provides local refinements, we adapt the design into a *global–local* structure: f_S is restricted to operate within each frame, while only f_D propagates global video context in a causal manner. More specifically, Equation (3.1) can be re-expressed as an autoregressive factorization over frames $\boldsymbol{x}_n$:

$$p_\theta(\boldsymbol{x}) = \prod_{n=1}^{N} p_\theta(\boldsymbol{x}_n \mid \boldsymbol{x}_{<n}) = \prod_{n=1}^{N} p_D(\boldsymbol{u}_n \mid \boldsymbol{u}_{<n})\left|\det J_{f_S}(\boldsymbol{x}_n)\right|, \tag{3.2}$$

where $\boldsymbol{u}_n = f_S(\boldsymbol{x}_n)$ denotes the local latents for frame $\boldsymbol{x}_n$. Here, the deep block is itself an autoregressive flow, capturing both intra-frame raster ordering and inter-frame causal dependencies.

Formulating STARFlow-V in a *global–local* manner (Equation (3.2)) yields several benefits:

(a) **Universality.** Equation (3.2) preserves the universal approximation guarantee of STARFlow (Gu et al., 2025): the local stack f_S still realizes per-pixel infinite Gaussian mixtures via alternating causal masks, so expressivity is not curtailed by restricting f_S to within-frame contexts.

(b) **Robustness.** Intuitively, Equation (3.2) can be viewed as a **continuous language model for videos**: the deep-flow term $p_D(\boldsymbol{u}_n \mid \boldsymbol{u}_{<n})$ acts as *Gaussian Next-Token Prediction* (cf. the affine form in Equation (2.2)) in latent space, while the shallow flow supplies the Jacobian factor $|\det J_{f_S}(\boldsymbol{x}_n)|$, yielding a flexible density over $\boldsymbol{x}$. Compared to modeling $\boldsymbol{x}$ directly (arbitrarily multimodal), the latent $\boldsymbol{u}$ is unimodal at each step, easier to regress, and more tolerant to small prediction errors. Crucially, the sampling phase via f_D^{-1} conditions on previously generated *latents* rather than pixels, so data-space errors do not propagate forward, mitigating the compounding error typical of autoregressive diffusion. Unlike diffusion-style noise conditioning (Ho et al., 2022b; Chen et al., 2024a), which compromises information to gain robustness and introduces extra parameters, our mappings $\boldsymbol{u} \leftrightarrow \boldsymbol{x}$ are invertible, avoiding information loss by construction.

(c) **End-to-End Training.** The whole model is still NF. Consequently, all parameters are trained by exact MLE via the change-of-variables objective—no per-step denoising schedule or surrogate loss—simplifying optimization and reducing train–test mismatch.

(d) **Streamable Generation.** At inference time, f_D^{-1} samples $\boldsymbol{u}_n$ causally (token-by-token, frame-by-frame), and f_S^{-1} decodes each frame independently given $\boldsymbol{u}_n$. This process enables causal video synthesis since later frames cannot influence earlier ones.

3.2 Revisiting Noise-Augmented Training

As observed by Zhai et al. (2024), injecting *small* noise into the data is crucial for stabilizing NF training. Concretely, we learn STARFlow-V on a σ-smoothed density $q_\sigma(\tilde{\boldsymbol{x}}) = (p * \mathcal{N}(0, \sigma^2 I))(\tilde{\boldsymbol{x}})$. A side effect is that

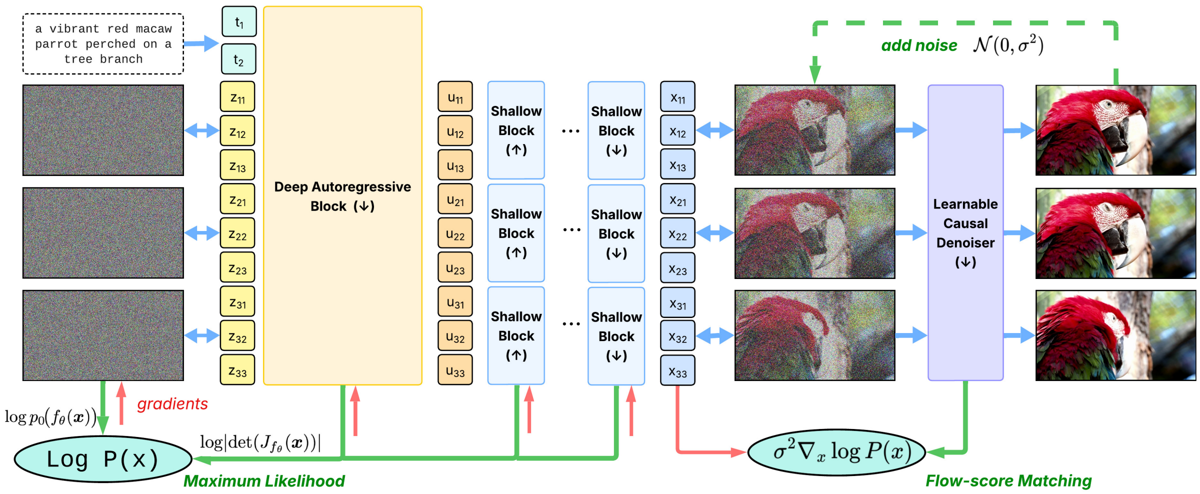


Figure 2 An illustrated pipeline of STARFlow-V which shows (1) the proposed global-local architecture; (2) joint training with the learnable denoiser with the proposed Flow-score Matching. During sampling, STARFlow-V takes the encoded text condition $\boldsymbol{t}$ and transforms the noise $\boldsymbol{z}$ through deep global block to intermediate features $\boldsymbol{u}$, followed by several local shallow blocks to produce a slightly noised video. Finally, a learnable causal denoiser refines this output into the final clean video $\boldsymbol{x}$.

the model naturally generates slightly noisy samples, necessitating a post-processing step to recover the clean ones. We first examined the existing options for this purpose:

(a) **Decoder Fine-tuning** We followed STARFlow (Gu et al., 2025), adopting their strategy of fine-tuning the VAE decoder to denoise noisy latents using a GAN objective (Rombach et al., 2022). However, our preliminary experiments suggest that this approach is not readily applicable to *3D causal* VAEs: under Gaussian-noised latent inputs, the decoder fails to maintain temporal consistency in the generated videos due to limited receptive fields.

(b) **Score-based Denoising** Instead of decoder fine-tuning, TARFlow (Zhai et al., 2024) proposes to denoise using the *learned flow* itself via score-based updates. For a noisy sample $\tilde{\boldsymbol{x}} \sim q_\sigma$, the continuity equation gives $\partial_\sigma \tilde{\boldsymbol{x}} = -\sigma \nabla_{\tilde{\boldsymbol{x}}} \log q_\sigma(\tilde{\boldsymbol{x}})$. So for sufficiently small σ, a single Euler step yields the Tweedie estimator:

$$\boldsymbol{x} \;\approx\; \tilde{\boldsymbol{x}} - \sigma\, \partial_\sigma \tilde{\boldsymbol{x}} \;=\; \tilde{\boldsymbol{x}} + \sigma^2 \nabla_{\tilde{\boldsymbol{x}}} \log q_\sigma(\tilde{\boldsymbol{x}}). \tag{3.3}$$

With normalizing flows, we replace q_σ by the learned density p_θ, and compute $\nabla_{\tilde{\boldsymbol{x}}} \log p_\theta(\tilde{\boldsymbol{x}})$ via automatic differentiation through the flow, which amounts to an additional forward–backward pass. However, this score-based denoising presents two issues: **(1) Noisy gradients.** The learned density p_θ is imperfect; its score $\nabla_{\tilde{\boldsymbol{x}}} \log p_\theta(\tilde{\boldsymbol{x}})$ often contains high-frequency noise, which manifests as bright speckle-like artifacts—especially in regions with large motion; **(2) Non-causality of the score.** Even if p_θ is modeled causally, the score $\nabla_{\tilde{\boldsymbol{x}}} \log p_\theta(\tilde{\boldsymbol{x}})$ is, by definition, global: the gradient at time n depends on likelihood terms involving future frames $m > n$. This breaks causality, undermining the promised streamable generation.

Proposed Approach: Flow-Score Matching To address these issues, we introduce a lightweight neural denoiser s_ϕ trained alongside the flow f_θ to regress the model's score:

$$\mathcal{L}_{\text{denoise}}(\phi) = \mathbb{E}_{\boldsymbol{x},\boldsymbol{\epsilon}} \big\| \, s_\phi(\tilde{\boldsymbol{x}}) \;-\; \sigma \, \nabla_{\tilde{\boldsymbol{x}}} \log p_\theta(\tilde{\boldsymbol{x}}) \, \big\|_2^2, \qquad \tilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\epsilon},\; \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 I). \tag{3.4}$$

At inference, we replace the raw score in the update (cf. Equation (3.3)) with the learned denoiser s_ϕ. This *flow-score matching* (FSM) is simple yet effective. First, the smooth inductive bias of neural networks suppresses stochastic high-frequency artifacts in $\nabla_{\tilde{\boldsymbol{x}}} \log p_\theta$. Second, we can encode causality directly in s_ϕ, re-ensuring streamable behavior. Concretely, we parameterize s_ϕ with a one–frame look-ahead while remaining globally causal (one-step latency)[1]. We approximate the score at step n by $s_\phi(\tilde{\boldsymbol{x}}_{\le n+1}) \;\approx\; \big(\sigma \, \nabla_{\tilde{\boldsymbol{x}}} \log p_\theta(\tilde{\boldsymbol{x}})\big)_n$.

[1]Strictly causal ($\le n$) fails as temporal *differences* are pivotal to determining the denoising direction.

Model	Total	Quality	Semantic	Aesthetic	Object	Multi Obj.	Human	Spatial	Scene
Closed-source models									
Gen-3 (Germanidis, 2024)	82.32	84.11	75.17	63.34	87.81	53.64	96.40	65.09	54.57
Veo3 (Google DeepMind, 2025)	85.06	85.70	82.49	63.81	93.89	82.20	99.40	84.26	57.43
Diffusion models									
OpenSora-v1.1 (Zheng et al., 2024)	75.66	77.74	67.36	50.12	86.76	40.97	84.20	52.47	38.63
CogVideoX (Yang et al., 2024)	80.91	82.18	75.83	60.82	83.37	62.63	98.00	69.90	51.14
HunyuanVideo (Kong et al., 2024)	83.24	85.09	75.82	60.36	86.10	68.55	94.40	68.68	53.88
Wan2.1-T2V (Wan et al., 2025)	83.69	85.59	76.11	66.07	86.28	69.58	95.40	75.39	45.75
Autoregressive (Diffusion) models									
CogVideo (Hong et al., 2022)	67.01	72.06	46.83	38.18	73.40	18.11	78.20	18.24	28.24
Emu3 (Wang et al., 2024b)	80.96	84.09	68.43	59.64	86.17	44.64	77.71	68.73	37.11
NOVA (Deng et al., 2024)	80.12	80.39	79.05	59.42	92.00	77.52	95.20	77.52	54.06
SkyReel-v2 (Chen et al., 2025)	83.90	84.70	80.80	-	-	-	-	-	-
MAGI-1-distill (Teng et al., 2025)	77.92	80.98	65.68	62.43	82.37	35.08	84.20	57.75	26.28
Normalizing Flows									
STARFlow-V (Ours)	78.67	80.24	72.37	54.48	86.65	53.48	94.00	49.84	47.08
STARFlow-V† (Ours)	79.70	80.76	75.43	59.73	80.61	56.04	98.13	76.08	48.21
STARFlow-V† (Ours, non-Causal)	79.22	80.34	74.71	58.70	81.08	54.60	98.40	73.15	49.61

Table 1 Text-to-video evaluation on VBench (Huang et al., 2024). The baseline data is from the leaderboard. Following Yang et al. (2025), we also evaluate with the official GPT-augmented prompts (Rewriter), with longer and more descriptive text inputs. † denotes results using Rewriter prompts.

Finally, we train s_ϕ jointly with f_θ at **minimal overhead**: since f_θ is trained by maximizing $\log p_\theta$, we cache the input gradients from the backward pass and reuse it as the target for s_ϕ.

3.3 Fast Inference

While STARFlow-V leverages parallel computation during training via causal masking, generation at inference time is carried out sequentially (one token at a time) through multiple AF blocks, which can be *extremely* computationally demanding for long video sequences. For instance, generating a 5s 480p video under 16 fps using a pre-trained 3B parameter model requires over 30 minutes, which is far from real-time performance. To enable fast inference, we introduce two strategies:

Block-wise Jacobi Iteration Rather than sampling continuous tokens strictly autoregressively, we accelerate inference by recasting inversion as solving a nonlinear fixed-point system with parallel solvers such as Jacobi iteration (Porsching, 1969; Kelley, 1995), a strategy recently used to speed up autoregressive models (Song et al., 2021; Teng et al., 2024; Liu and Qin, 2025; Zhang et al., 2025). Specifically, the inverse of Equation (2.2) can be written as the fixed-point equation

$$\boldsymbol{x} \;=\; \mu_\theta(\boldsymbol{x}\odot\boldsymbol{m}) \;+\; \sigma_\theta(\boldsymbol{x}\odot\boldsymbol{m})\cdot\boldsymbol{z}, \tag{3.5}$$

where $\boldsymbol{m}$ is a (self-exclusive) causal mask. This induces a *triangular* system that admits convergence under nonlinear Jacobi iteration (Saad, 2003): starting from an initial sequence estimate $\boldsymbol{x}^{(0)}$, iterate $\boldsymbol{x}^{(k+1)} = \mu_\theta(\boldsymbol{x}^{(k)}\odot\boldsymbol{m}) + \sigma_\theta(\boldsymbol{x}^{(k)}\odot\boldsymbol{m})\cdot\boldsymbol{z}$ until a converge criterion is satisfied. We monitor a scale-normalized residual, $\|\boldsymbol{x}^{(k+1)}-\boldsymbol{x}^{(k)}\|_2^2/\|\boldsymbol{x}^{(k+1)}\|_2^2 < \tau$ with $\tau = 0.001$ by default. Although the worst-case iteration count scales with sequence length (*e.g.*, near-Markovian process), video generation exhibits strong global structure, substantially accelerating convergence in practice. The procedure is also *guidance-compatible*, as proposed in (Gu et al., 2025), which involves computing the guided parameters $\hat{\mu}$ and $\hat{\sigma}$ and then substituting them.

To further accelerate sampling, we adopt a block-wise Jacobi scheme in the spirit of Song et al. (2021); Liu and Qin (2025). The token sequence is partitioned into contiguous blocks of size B, which are processed sequentially across blocks but in parallel within each block. Within each block we run the Jacobi updates, while states from completed blocks are cached as context (*e.g.*, KV cache) for subsequent blocks—analogous to standard AR inference. We also apply a video-aware initialization: for a new frame, the initial estimate $\boldsymbol{x}_{n+1}^{(0)}$ is initialized from the previously converged frame $\boldsymbol{x}_n^{(k)}$. Overall, we adopt block-based iteration within each AF block, yielding $\approx$ **15**$\times$ lower inference latency relative to standard autoregressive decoding, while preserving visual fidelity.

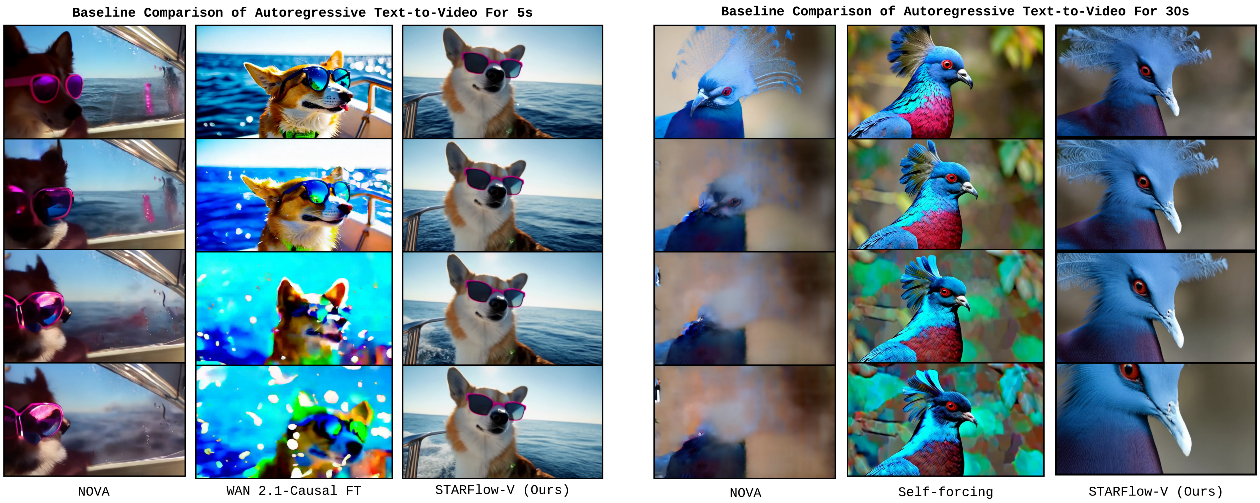


Figure 3 STARFlow-V comparison against baselines on autoregressive generation for both trained length (5s) and long-horizon generation (30s). **Please refer to more video comparison in the project page.**

Pipelined Decoding As described in Section 3.1, the global–local design applies standard global left-to-right autoregression in the deep block f_D, while the shallow blocks f_S traverse each frame independently. This enables a pipelined schedule (analogous to pipeline parallelism (Huang et al., 2019)): f_D runs continuously without waiting on f_S, and, in parallel, f_S threads consume f_D's outputs, immediately refine them, and then denoise. Because f_D is typically the slowest stage, end-to-end latency is dominated by the deep block.

3.4 Versatility Across Tasks

STARFlow-V can be trained for different video generation tasks. By default, STARFlow-V is trained for text-to-video generation on large-scale text–video pairs. Without modifying the backbone, we support the following settings:

(a) **Image-to-Video Generation.** We directly treat the first frame as observed conditioning. Owing to the invertiblity, *no separate encoder is required*: we encode the observed frame via the flow forward to initialize the KV cache; subsequent frames are then generated.

(b) **Video-to-Video Generation.** Given a source clip $\boldsymbol{x}_{0:T}$, we treat all frames as observed conditioning and—thanks to invertibility—use the same backbone to flow-encode them and populate the KV cache. The model then autoregressively rolls out the target clip $\hat{\boldsymbol{x}}_{0:T}$ under optional task cues (e.g., in/outpainting masks, edit text, camera/pose), copying through unedited regions while synthesizing edits. This mirrors our image-to-video path but operates framewise over the whole clip without a separate encoder.

(c) **Longer Generation.** Our model generates videos far longer than those seen during training via a sliding-window (chunk-to-chunk) schedule in the deep block. After producing a latent chunk $\boldsymbol{u}$, we warm-start the next step by rebuilding the KV cache: we re-run f_D on the last Δ latents (the overlap) and then continue autoregression to synthesize the next $N - \Delta$ latents. f_S then process the latents per frame, enabling streaming output. To mitigate boundary mismatch, we randomly drop the first frame during training to simulate restart.

4 Experiments

4.1 Experimental Setup

Datasets. We construct a diverse and high-quality collection of video datasets to train STARFlow-V. Specifically, we leverage the high-quality subset of Panda (Chen et al., 2024b) mixed with an in-house stock video dataset, with a total number of 70M text-video pairs. For all videos, we keep their raw captions, and apply a video captioner (Wang et al., 2024a) to generate a longer description to cover the details. The ratio of training

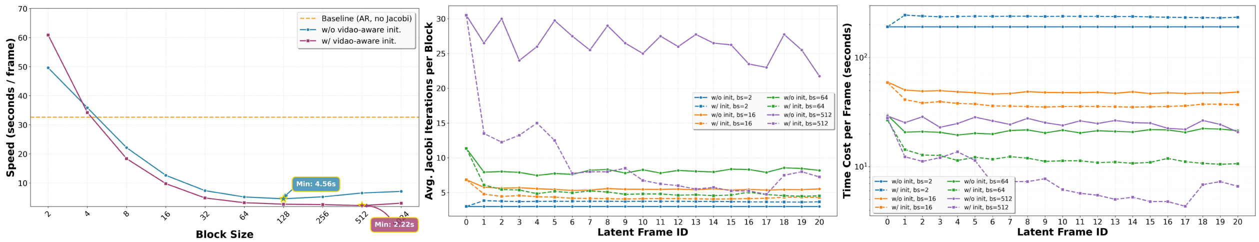


Figure 4 Comparison between speed and block size in block-wise Jacobi iteration.

using raw and synthetic captions during training is 1 : 9. Besides, following previous works (Lin et al., 2024), we additionally include 400M text-image pairs for joint training. To support video-to-video generation and editing, we additionally finetune the pretrained STARFlow-V on the Señorita (Zi et al., 2025), a large-scale and high-quality instruction-based video editing dataset spanning 18 well-defined editing subcategories.

Evaluation. We perform both quantitative and qualitative evaluations on STARFlow-V, and compare against baselines using VBench (Huang et al., 2024), which benchmarks text-to-video generation across 16 dimensions, including quality, semantics, temporal consistency, and spatial reasoning.

Model and Training Details. We adopt the 3D Causal VAE from WAN2.2 (Wan et al., 2025), which compresses spatial dimensions by $\times16$ and the temporal dimension by $\times4$ into a 48-channel latent space. We train progressively: we initialize from an image (single-frame) model, then scale to a 7B-parameter video model by increasing the deep-block capacity. For resolution, we use a curriculum from 384p to 480p while keeping the sequence length fixed at 81 frames. For the learnable denoiser, we used a 8-layer Transformer with the same channel dimension as shallow block. We include more implementation details in Appendix.

Baselines. We compare with three baselines: (i) **WAN-2.1 Causal**, the autoregressive variant of WAN (Wan et al., 2025) finetuned with the CausVid strategy (Yin et al., 2025); (ii) **Self-Forcing** (Huang et al., 2025), finetuned from WAN-2.1 Causal-FT to mitigate train–test mismatch; and (iii) **NOVA**(Deng et al., 2024), a native autoregressive diffusion model that does not rely on vector quantization. The orginal model predicts in a chunk-based fashion. For fair comparisons, we also execute results in the pure AR settings. Besides, we also report quantitative results on VBench with official scores.

4.2 Quantitative Results

Table 1 reports T2V results on VBench (Huang et al., 2024). While STARFlow-V does not yet match the strongest diffusion-based video generators, it attains performance in the same range as recent causal diffusion baselines, substantially narrowing the historical gap between NFs and diffusion models for video. To the best of our knowledge, STARFlow-V is the **first NF-based text-to-video model** to reach this level of quality, indicating that NFs can be a viable alternative when invertibility and exact likelihood (as shown in (Zhai et al., 2024)) are desired. We also include a variant trained without local constraints; its VBench scores remain very close to the causal version, indicating that enforcing causal structure does not incur a noticeable loss in perceptual quality.

4.3 Qualitative Results

T2V & I2V Tasks As illustrated in Figure 1, STARFlow-V naturally supports both T2V and I2V generation. The examples show that STARFlow-V produces temporally smooth and visually faithful sequences in both settings. Importantly, both T2V and I2V results are obtained from the *same* model without additional tuning: thanks to invertibility and causal modeling, the decoder can be reused as an encoder when a conditioning image is provided.

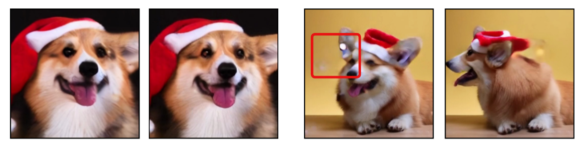
V2V Tasks As shown in Figure 1, STARFlow-V handles diverse V2V tasks from object-level to dense prediction within a single framework simply by changing the instruction. These results illustrate the potential of using our NF-based model for general video editing and reasoning.

Against Autoregressive Diffusion Models In Figure 3, we compare STARFlow-V with two representative autoregressive diffusion models. For the dog-with-sunglasses example, NOVA (Deng et al., 2024) exhibits gradual blurring and loss of identity, while WAN 2.1-Causal FT shows strong artifacts and color distortions. In contrast, STARFlow-V maintains clean, sharp, and temporally consistent frames, indicating stronger robustness to exposure bias. The right block of Figure 3 further shows that STARFlow-V sustains stable, coherent generations when extended to 30 seconds—well beyond its 5-second training horizon—where NOVA (Deng et al., 2024) and Self-Forcing (Huang et al., 2025) suffer from blurring, color drift, and structural deformation. We further report **quantitative** metrics for evaluating drifting effects across baselines and our model in the Appendix.

4.4 Ablation Study

Choice of Denoiser Figure 5 provides an ablation on the denoiser design. As shown in the top row, Decoder-finetuning (Gu et al., 2025) tends to lose temporal consistency with noticeable frame-to-frame jitter, while score-based denoising (Zhai et al., 2024) introduces bright speckle artifacts, especially in regions of large motion. The quantitative comparison (bottom) further shows that our proposed flow–score matching achieves substantially better video reconstruction under latent-space noise injection, outperforming both alternatives by a clear margin.

Hyper-parameters of Block-wise Jacobi Iteration We analyze how the block size used in the block-wise Jacobi Iteration influences the runtime of the deep block. As shown in Figure 4 (left), the runtime initially decreases as the block size increases, reflecting better utilization of intra-block parallelism, but then rises slightly again when the block size becomes too large. This trend suggests a trade-off: while larger block sizes increase parallelism, excessively large blocks requires more iterations within each block to achieve convergence.



(a) Decoder-finetuning (b) Score-based Denoising

Method	PSNR↑	SSIM↑
No noise	32.22	0.8907
Decoder fine-tuning (Gu et al., 2025)	23.95	0.6403
Score-based denoising (Zhai et al., 2024)	22.05	0.6490
Flow-score matching (ours)	**26.69**	**0.7601**

Figure 5 Ablation study for the choice of denoiser. We compare video VAE reconstruction quality across denoising approaches over $1,000$ random videos with large motions.

We also examine the impact of video-aware initialization on runtime. As illustrated in Figure 4 (left), initializing the first Jacobi iteration of each frame using the converged state from the previous frame substantially reduces runtime across almost all block sizes except for small block sizes. This improvement likely stems from the strong temporal coherence present in natural videos, where neighboring frames provide effective warm starts that appear to facilitate faster iterative updates. Overall, video-aware initialization leads to observed improvements across block sizes.

We further analyze the runtime breakdown across latent frames in Figure 4 (right). Video-aware initialization yields the largest gains for large block sizes after the first frame, where convergence would otherwise require many more inner steps. Based on this observation, we adopt an asymmetric default strategy: *use a medium block size (e.g., 64) for the first frame, and a larger block size (e.g., 512) for subsequent frames with video-aware initialization.*

5 Conclusion and Limitations

We presented STARFlow-V, an end-to-end video generative model based on autoregressive normalizing flows. As shown experimentally, STARFlow-V delivers strong long-horizon coherence and fine-grained controllability across text-to-video, image-to-video and video-to-video tasks, and shows consistent gains over autoregressive diffusion baselines at 480p/81f. As a bonus, STARFlow-V can be used natively for likelihood estimation.

While the results are encouraging, there are still limitations to overcome. (1) *Latency.* Despite the proposed accelerated sampling, inference remains far from real time on commodity GPUs. (2) *Data quality and scaling.* Progress is bounded by dataset noise and bias; we do not observe a clean scaling law under current curation.

(3) *Non-physical generation.* Due to the current model scale and available data, we still observe many unrealistic, non-physical generations (see Figure 6), such as an octopus passing through the wall of a jar and a rock spontaneously appearing beneath a goat just as it lands.

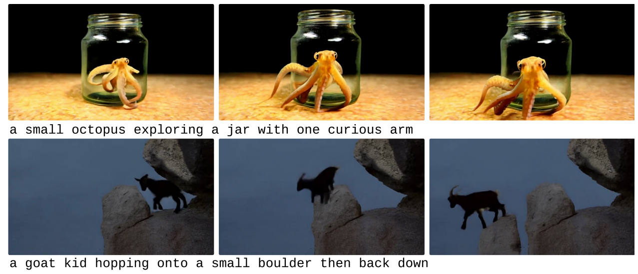


Figure 6 Failure cases of generation from STARFlow-V.

Looking forward, we see several promising directions. First, we aim to reduce generation latency, for example through more efficient sampling schedules and architectural optimizations. Second, we plan to study distillation and pruning to obtain compact student models that retain most of the performance of the full system. Third, we will revisit dataset curation and active data selection, with a particular focus on challenging, large-motion sequences and physically grounded scenarios; this is crucial for improving physical plausibility, reducing non-physical failure cases, and enabling clearer scaling behavior at higher fidelity.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H. Campbell, and Sergey Levine. Stochastic variational video prediction. In *International Conference on Learning Representations (ICLR)*, 2018. doi: 10.48550/arXiv.1710.11252. URL https://openreview.net/forum?id=rk49Mg-CW.

Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv e-prints*, pages arXiv–2506, 2025.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. URL https://openai.com/research/video-generation-models-as-world-simulators.

Lluis Castrejon, Nicolas Ballas, and Aaron Courville. Improved conditional vrnns for video prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7608–7617, 2019.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024a.

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

Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang-wei Chao, Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, et al. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13320–13331, 2024b.

Google DeepMind. Veo 3: Ai video generator with audio. https://deepmind.google/models/veo/, 2025. Accessed: 2025-08-25.

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. *arXiv preprint arXiv:2412.14169*, 2024.

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.

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

Anastasis Germanidis. Introducing gen-3 alpha: A new frontier for video generation, 2024.

Google DeepMind. Genie 2: A large-scale foundation world model. https://deepmind.google/discover/blog/genie-2-a-large-scale-foundation-world-model/, 2024. Blog.

Google DeepMind. Veo 3 Technical Report. https://storage.googleapis.com/deepmind-media/veo/Veo-3-Tech-Report.pdf, 2025. Accessed: Sep 24, 2025.

Jiatao Gu, Tianrong Chen, David Berthelot, Huangjie Zheng, Yuyang Wang, Ruixiang Zhang, Laurent Dinh, Miguel Angel Bautista, Josh Susskind, and Shuangfei Zhai. Starflow: Scaling latent normalizing flows for high-resolution image synthesis. *arXiv preprint arXiv:2506.06276*, 2025.

David Ha and Jurgen Schmidhuber. World models. *NeurIPS*, 2018. doi: 10.1007/bfb0007224.

Danijar Hafner et al. Mastering diverse control tasks through world models. *Nature*, 2025. Also available as arXiv:2301.04104 (DreamerV3).

Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In *International conference on machine learning*, pages 2722–2730. PMLR, 2019.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022a.

Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.*, 23:47–1, 2022b.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems*, 35:8633–8646, 2022c.

Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. Gaia-1: A generative world model for autonomous driving. *arXiv preprint arXiv:2309.17080*, 2023.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025.

Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Dehao Chen, Mia Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V Le, Yonghui Wu, et al. Gpipe: Efficient training of giant neural networks using pipeline parallelism. *Advances in neural information processing systems*, 32, 2019.

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

Carl T Kelley. *Iterative methods for linear and nonlinear equations*. SIAM, 1995.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, 29, 2016.

Dan Kondratyuk, Lijun Yu, Xiuye Gu, Jose Lezama, Jonathan Huang, Grant Schindler, Rachel Hornung, Vighnesh Birodkar, Jimmy Yan, Ming-Chang Chiu, et al. Videopoet: A large language model for zero-shot video generation. In *International Conference on Machine Learning*, pages 25105–25124. PMLR, 2024.

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

Manoj Kumar, Mohammad Babaeizadeh, Dumitru Erhan, Chelsea Finn, Sergey Levine, Laurent Dinh, and Durk Kingma. Videoflow: A conditional flow-based model for stochastic video generation. *arXiv preprint arXiv:1903.01434*, 2019.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *arXiv preprint arXiv:2406.11838*, 2024.

Bin Lin, Yunyang Ge, Xinhua Cheng, Zongjian Li, Bin Zhu, Shaodong Wang, Xianyi He, Yang Ye, Shenghai Yuan, Liuhan Chen, et al. Open-sora plan: Open-source large video generation model. *arXiv preprint arXiv:2412.00131*, 2024.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Ben Liu and Zhen Qin. Accelerate tarflow sampling with gs-jacobi iteration. *arXiv preprint arXiv:2505.12849*, 2025.

OpenAI. Gpt-4o system card. https://openai.com/index/gpt-4o-system-card/, 2024a. Accessed: April 12, 2025.

OpenAI. Video generation models as world simulators. https://openai.com/index/video-generation-models-as-world-simulators/, 2024b.

George Papamakarios, Iain Murray, and Theo Pavlakou. Masked autoregressive flow for density estimation. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 2338–2347, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/6c1da886822c67822bcf3679d04369fa-Abstract.html.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

TA Porsching. Jacobi and gauss–seidel methods for nonlinear network problems. *SIAM Journal on Numerical Analysis*, 6(3):437–449, 1969.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/rezende15.html.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

Yousef Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.

Team Seawead, Ceyuan Yang, Zhijie Lin, Yang Zhao, Shanchuan Lin, Zhibei Ma, Haoyuan Guo, Hao Chen, Lu Qi, Sen Wang, et al. Seaweed-7b: Cost-effective training of video generation foundation model. *arXiv preprint arXiv:2504.08685*, 2025.

Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3626–3636, 2022.

Kiwhan Song, Boyuan Chen, Max Simchowitz, Yilun Du, Russ Tedrake, and Vincent Sitzmann. History-guided video diffusion. *arXiv preprint arXiv:2502.06764*, 2025.

Yang Song, Chenlin Meng, Renjie Liao, and Stefano Ermon. Accelerating feedforward computation via parallel nonlinear equation solving. In *International Conference on Machine Learning*, pages 9791–9800. PMLR, 2021.

Hansi Teng, Hongyu Jia, Lei Sun, Lingzhi Li, Maolin Li, Mingqiu Tang, Shuai Han, Tianning Zhang, WQ Zhang, Weifeng Luo, et al. Magi-1: Autoregressive video generation at scale. *arXiv preprint arXiv:2505.13211*, 2025.

Yao Teng, Han Shi, Xian Liu, Xuefei Ning, Guohao Dai, Yu Wang, Zhenguo Li, and Xihui Liu. Accelerating autoregressive text-to-image generation with training-free speculative jacobi decoding. *arXiv preprint arXiv:2410.01699*, 2024.

Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1526–1535, June 2018.

Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Advances in Neural Information Processing Systems*, volume 29, 2016. doi: 10.48550/arXiv.1609.02612. URL https://papers.nips.cc/paper_files/paper/2016/hash/04025959b191f8f9de3f924f0940515f-Abstract.html.

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

Jiawei Wang, Liping Yuan, Yuchen Zhang, and Haomiao Sun. Tarsier: Recipes for training and evaluating large video description models. *arXiv preprint arXiv:2407.00634*, 2024a.

Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024b.

Bohan Wu, Suraj Nair, Roberto Martín-Martín, Li Fei-Fei, and Chelsea Finn. Greedy hierarchical variational autoencoders for large-scale video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2318–2328, June 2021.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.

Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*, 2021. doi: 10.48550/arXiv.2104.10157.

Sherry Yang, Yilun Du, Seyed Kamyar Seyed Ghasemipour, Jonathan Tompson, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *NeurIPS 2023 Workshop on Generalization in Planning*, 2023.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *The Thirteenth International Conference on Learning Representations*, 2025.

Zixuan Ye, Huijuan Huang, Xintao Wang, Pengfei Wan, Di Zhang, and Wenhan Luo. Stylemaster: Stylize your video with artistic generation and translation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2630–2640, 2025.

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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22963–22974, 2025.

Lijun Yu, José Lezama, Nitesh Bharadwaj Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G Hauptmann, et al. Language model beats diffusion-tokenizer is key to visual generation. In *ICLR*, 2024.

Shenghai Yuan, Jinfa Huang, Xianyi He, Yunyang Ge, Yujun Shi, Liuhan Chen, Jiebo Luo, and Li Yuan. Identity-preserving text-to-video generation by frequency decomposition. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12978–12988, 2025.

Shuangfei Zhai, Ruixiang ZHANG, Preetum Nakkiran, David Berthelot, Jiatao Gu, Huangjie Zheng, Tianrong Chen, Miguel Ángel Bautista, Navdeep Jaitly, and Joshua M Susskind. Normalizing flows are capable generative models. In *Forty-second International Conference on Machine Learning*.

Shuangfei Zhai, Ruixiang Zhang, Preetum Nakkiran, David Berthelot, Jiatao Gu, Huangjie Zheng, Tianrong Chen, Miguel Angel Bautista, Navdeep Jaitly, and Josh Susskind. Normalizing flows are capable generative models. *arXiv preprint arXiv:2412.06329*, 2024.

Jiaru Zhang, Juanwu Lu, Ziran Wang, and Ruqi Zhang. Inference acceleration of autoregressive normalizing flows by selective jacobi decoding. *arXiv preprint arXiv:2505.24791*, 2025.

Guangting Zheng, Qinyu Zhao, Tao Yang, Fei Xiao, Zhijie Lin, Jie Wu, Jiajun Deng, Yanyong Zhang, and Rui Zhu. Farmer: Flow autoregressive transformer over pixels. *arXiv preprint arXiv:2510.23588*, 2025.

Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*, 2024.

Bojia Zi, Penghui Ruan, Marco Chen, Xianbiao Qi, Shaozhe Hao, Shihao Zhao, Youze Huang, Bin Liang, Rong Xiao, and Kam-Fai Wong. Se\˜ norita-2m: A high-quality instruction-based dataset for general video editing by video specialists. *arXiv preprint arXiv:2502.06734*, 2025.

A Derivations and Algorithms

A.1 Derivation of STARFlow-V.

(1) Why an autoregressive Gaussian model in $\boldsymbol{u}$ is a normalizing flow. Let $T_\theta : \boldsymbol{u} \mapsto \boldsymbol{z}$ be the *triangular* autoregressive map applied by the deep block f_D (within a frame and across frames in the global order). For token index i in that order,

$$\boldsymbol{z}_i \;=\; \frac{\boldsymbol{u}_i - \mu_\theta(\boldsymbol{u}_{<i})}{\sigma_\theta(\boldsymbol{u}_{<i})}, \qquad \sigma_\theta(\cdot) > 0, \tag{A.1}$$

with inverse

$$\boldsymbol{u}_i \;=\; \sigma_\theta(\boldsymbol{u}_{<i})\,\boldsymbol{z}_i \;+\; \mu_\theta(\boldsymbol{u}_{<i}). \tag{A.2}$$

Because each $\boldsymbol{z}_i$ depends only on $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_i)$ and $\sigma_\theta > 0$, T_θ is bijective and continuously differentiable. The Jacobian is lower triangular with diagonal entries $\partial \boldsymbol{z}_i / \partial \boldsymbol{u}_i = 1/\sigma_\theta(\boldsymbol{u}_{<i})$, thus

$$\log\big|\det J_{T_\theta}(\boldsymbol{u})\big| = -\sum_i \log \sigma_\theta(\boldsymbol{u}_{<i}). \tag{A.3}$$

With a standard normal prior $p_0(\boldsymbol{z}) = \prod_i \mathcal{N}(\boldsymbol{z}_i; 0, I)$,

$$\log p_D(\boldsymbol{u}) = \log p_0\big(T_\theta(\boldsymbol{u})\big) + \log\big|\det J_{T_\theta}(\boldsymbol{u})\big| = -\tfrac{1}{2}\sum_i \boldsymbol{z}_i^2 \;-\; \sum_i \log \sigma_\theta(\boldsymbol{u}_{<i}) \;+\; \text{const}, \tag{A.4}$$

which is essentially the regression objective through maximum likelihood estimation over $\boldsymbol{u}$. Therefore, the deep block realizes a valid normalizing flow. Composing with the shallow block gives $f_\theta = f_D \circ f_S$ and yields the data density in Equation (3.1).

(2) How we get the autoregressive distribution. From the global–local factorization (Equation (3.2)),

$$p_\theta(\boldsymbol{x}) = \prod_{n=1}^{N} p_D(\boldsymbol{u}_n \mid \boldsymbol{u}_{<n}) \, \big|\det J_{f_S}(\boldsymbol{x}_n)\big|, \qquad \boldsymbol{u}_n = f_S(\boldsymbol{x}_n). \tag{A.5}$$

Within a frame n, index tokens $k = 1, \ldots, HW \cdot D$ in raster (or block) order and we have Equation (A.4) which models p_D as Gaussian. The shallow-block contributes the additional log–det $\sum_n \log|\det J_{f_S}(\boldsymbol{x}_n)|$, forming an expressive distribution.

(3) Noise & denoising: what the model looks like. Following the noise-augmented training (§3.2), let $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \sigma\epsilon$, $\epsilon \sim \mathcal{N}(0, I)$. The Tweedie single-step denoiser in the flow setting (Equation (3.3)) suggests the update $\boldsymbol{x} \approx \tilde{\boldsymbol{x}} + \sigma^2 \nabla_{\tilde{\boldsymbol{x}}} \log p_\theta(\tilde{\boldsymbol{x}})$. To avoid high-frequency artifacts and to preserve streamability, we fit a *causal* denoiser s_ϕ via flow-score matching (Equation (3.4)) and then use

$$\hat{\boldsymbol{x}} \;=\; \tilde{\boldsymbol{x}} \;+\; \sigma\, s_\phi(\tilde{\boldsymbol{x}}) \;\;\approx\;\; \tilde{\boldsymbol{x}} \;+\; \sigma^2 \nabla_{\tilde{\boldsymbol{x}}} \log p_\theta(\tilde{\boldsymbol{x}}), \tag{A.6}$$

where s_ϕ uses a block-causal mask with at most one-frame look-ahead to retain strict streamability.

Algorithm 1 Training STARFlow-V with noise augmentation and flow-score matching

Require: video dataset $\mathcal{D}$; noise level σ; FSM weight λ_{den}

repeat
  Sample mini-batch $\boldsymbol{x} \sim \mathcal{D}$ and noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$
  Noise-augment: $\tilde{\boldsymbol{x}} \leftarrow \boldsymbol{x} + \sigma\, \boldsymbol{\epsilon}$ ▷ as in §3.2
  Shallow forward: $\boldsymbol{u} \leftarrow f_S(\tilde{\boldsymbol{x}})$ ▷ alternating masked AF blocks, within-frame
  Deep forward: $\boldsymbol{z} \leftarrow f_D(\boldsymbol{u})$ ▷ causal Transformer AF over global order
  Standard NF NLL: $\mathcal{L}_{\text{NLL}}(\theta) \leftarrow -\big[\log p_0(\boldsymbol{z}) + \log|\det J_{f_D}(\boldsymbol{u})||\det J_{f_S}(\tilde{\boldsymbol{x}})|\big]$
  Score target (stop-grad): $\mathbf{t} \leftarrow \sigma\, \nabla_{\tilde{\boldsymbol{x}}} \log p_\theta(\tilde{\boldsymbol{x}})$ ▷ reuse backward pass of $\mathcal{L}_{\text{NLL}}$; detach
  Flow-score Matching: $\mathcal{L}_{\text{FSM}}(\phi) \leftarrow \|\, s_\phi(\tilde{\boldsymbol{x}}) - \mathbf{t}\, \|_2^2$
  Total loss: $\mathcal{L} \leftarrow \mathcal{L}_{\text{NLL}}(\theta) + \lambda_{\text{den}}\, \mathcal{L}_{\text{FSM}}(\phi)$
  Update: $(\theta, \phi) \leftarrow (\theta, \phi) - \eta\, \nabla \mathcal{L}$
until convergence

Algorithm 2 Autoregressive sampling ($\boldsymbol{z} \to \boldsymbol{u} \to \boldsymbol{x}$)

Require: length N (frames or tokens), base prior $p_0(\boldsymbol{z}) = \mathcal{N}(0, I)$, shallow inverse f_S^{-1}, deep inverse f_D^{-1}, token order $\prec$

Sample $\boldsymbol{z} \sim \mathcal{N}(0, I)$ with the target shape
Initialize an empty latent sequence $\boldsymbol{u}$
for each element i in global order $\prec$ **do** ▷ causal AR over frames and within-frame tokens
  Compute (μ_i, σ_i): $(\mu_i, \sigma_i) \leftarrow f_D\big(\boldsymbol{u}_{<i}\big)$
  Invert deep at position i: $\boldsymbol{u}_i \leftarrow \sigma_i\, \boldsymbol{z}_i + \mu_i$ ▷ f_D^{-1}, triangular
end for
Invert shallow block: $\boldsymbol{x} \leftarrow f_S^{-1}(\boldsymbol{u})$
(One-step corrector) $\boldsymbol{x} \leftarrow \boldsymbol{x} + \sigma_{\text{test}}\, s_\phi(\boldsymbol{x})$
return $\boldsymbol{x}$

A.2 Training

Algorithm 1 shows the training algorithm of STARFlow-V for both the flow and the learnable denoiser.

A.3 Inference

Remarks. (i) When the deep map is sufficiently contractive in $\boldsymbol{u}$ (e.g., via scale clamping), the Jacobi iteration converges rapidly and enables wide parallelism within each block B. (ii) A common choice for $\mathcal{B}$ is to use spatial tiles per frame (no intra-tile dependencies) or even/odd raster groups, preserving the block-causal mask used in training.

B Implementation Details

B.1 Architecture Design

	3B	7B
Params	∼3B	∼7B
f_D width	3072	4096
f_S	identical (alt. masked AF; width d_S, depth L_S)	
Denoiser s_ϕ	8-layer Transformer, block-causal mask	
Init	from scratch	finetune from 3B

Table 2 Minimal comparison. Only f_D width differs; f_S and s_ϕ are unchanged.

3B. Same size as STARFlow but for *video*. The deep block f_D uses width 3072 (depth L_D, heads H_D). The

Algorithm 3 Jacobi-style parallel inversion of the deep autoregressive block

Require: base latent $\boldsymbol{z}$; initial guess $\boldsymbol{u}^{(0)}$ (e.g., zeros); block partition $\mathcal{B} = \{B_1, \ldots, B_J\}$ (non-overlapping, block-causal, $|B_j| = 4|B_1|$ for all block $j > 1$); max iters T; Frame size F; tolerance τ

for $j = 1, 2, \ldots, J$ **do**
 $[a, b] \leftarrow B_j$ ▷ indices of the j-th block
 if $j = 1$ **and** $a > F$ **then**
  Initialize $\boldsymbol{u}_{a:b} \leftarrow \boldsymbol{u}^{(0)}_{a:b}$ ▷ random initialization
 else
  Initialize $\boldsymbol{u}_{a:b} \leftarrow \boldsymbol{u}_{a-F:b-F}$ ▷ initialization from past frame
 end if
 repeat
  $t \leftarrow t + 1$
  for all $i \in B_j$ **in parallel do**
   $(\mu_i^{(t)}, \sigma_i^{(t)}) \leftarrow f_D\big(\boldsymbol{u}^{(t)}_{<i}\big)$
   $\boldsymbol{u}_i^{(t+1)} \leftarrow \sigma_i^{(t)}\, \boldsymbol{z}_i + \mu_i^{(t)}$
  end for
 until $\frac{\|\boldsymbol{u}^{(t+1)} - \boldsymbol{u}^{(t)}\|_2}{\|\boldsymbol{u}^{(t)}\|_2 + \varepsilon} \leq \tau$ **or** $t = T$
 $\boldsymbol{u}_{a:b} \leftarrow \boldsymbol{u}^{(t)}_{a:b}$
end for
Shallow inverse: $\boldsymbol{x} \leftarrow f_S^{-1}\big(\boldsymbol{u}^{(t+1)}\big)$
(One-step corrector) $\boldsymbol{x} \leftarrow \boldsymbol{x} + \sigma_{\text{test}}\, s_\phi(\boldsymbol{x})$
return $\boldsymbol{x}$

shallow stack f_S (alternating masked affine flows) and the denoiser s_ϕ (8-layer Transformer with block-causal mask) follow the standard design.

7B. Initialized from the 3B checkpoint and *only* widens the deep block f_D channels from 3072 to 4096. The shallow stack f_S and denoiser s_ϕ remain identical (same depths, heads, and widths).

B.2 Training Details

STARFlow-V is trained on 96 H100 GPUs using approximately 20 million videos. In all the experiments, we share the following training configuration for our proposed STARFlow-V.

```
training config:
    batch_size=96
    optimizer='AdamW'
    adam_beta1=0.9
    adam_beta2=0.95
    adam_eps=1e-8
    learning_rate=5e-5
    min_learning_rate=1e-6
    learning_rate_schedule=cosine
    weight_decay=1e-4
    mixed_precision_training=bf16
```

Progressive Video Training We adopt a progressive multi-stage training paradigm that gradually increases model size, resolution, and temporal horizon for stable and effective optimization.

- **3B Text-to-Image Training:** We initialize a 3B text-to-image model from the pretrained StarFlow (Gu et al., 2025), establishing a strong visual–textual backbone before introducing temporal modeling.
- **3B Image-Video Joint Training (384P, 45 frames):** The 3B model is then jointly trained on low-resolution images and videos at 384P. Each training clip contains 45 frames sampled at 16 fps, enabling the model to acquire short-term temporal dynamics.

Algorithm 4 Streaming long-sequence generation via *re-encode with forward*

Require: target length T (frames), window size W ($W \ll T$); deep inverse f_D^{-1}; shallow inverse f_S^{-1}; shallow forward f_S; deep forward f_D; prior $p_0(\boldsymbol{z})$

Initialize caches $\mathsf{KV} \leftarrow \varnothing$, latent buffer $\mathsf{U} \leftarrow \varnothing$
for $t = 1$ to T **do**
  Sample base: $\boldsymbol{z}_t \sim \mathcal{N}(0, I)$ for the next frame (or token block)
  Deep inverse: using cached state, compute $\boldsymbol{u}_t \leftarrow f_D^{-1}(\boldsymbol{z}_t\,; \mathsf{KV})$ and update the KV cache.
  Shallow inverse: $\boldsymbol{x}_t \leftarrow f_S^{-1}(\boldsymbol{u}_t)$
  Emit $\boldsymbol{x}_t$
  Re-encode (forward): $\hat{\boldsymbol{u}}_t \leftarrow f_S(\boldsymbol{x}_t)$ ▷ brings the produced frame back to U-space
  Update deep state: run f_D *forward* on $\hat{\boldsymbol{u}}_t$ to refresh KV (no sampling): $\_ \leftarrow f_D(\hat{\boldsymbol{u}}_t; \mathsf{KV})$
  Maintain sliding window: push $\hat{\boldsymbol{u}}_t$ into buffer U; if $|\mathsf{U}| > W$ pop the oldest
end for
return $\{\boldsymbol{x}_t\}_{t=1}^{T}$

- **7B Image-Video Joint Training (384P, 81 frames):** We expand the model to 7B parameters and continue joint training at 384P, doubling the temporal horizon from 45 to 81 frames to strengthen long-range temporal reasoning.
- **7B Image-Video Joint Training (480P, 81 frames):** Finally, we train the 7B model on higher-resolution 480P images and videos while maintaining the 81-frame temporal window.

Mixed-Resolution Training STARFlow-V is designed to support *mixed-resolution* inputs, allowing each frame to retain its native aspect ratio and spatial resolution. Similar to Gu et al. (2025), we assign each video sequence to one of nine predefined aspect-ratio bins, since all frames within a video share the same ratio. The pre-defined bins are 21:9, 16:9, 3:2, 5:4, 1:1, 4:5, 2:3, 9:16, and 9:21. To make the model explicitly aware of these visual formats, we incorporate both the fps and aspect-ratio tag into the text caption:

```
A video with {fps} fps:
{original_caption}
in a {aspect_ratio} aspect ratio.
```

Gradient Control We monitor the gradient norm throughout training to ensure stability. Specifically, to prevent gradient explosion, we enable gradient skipping after the first 100 steps: if the gradient norm exceeds a threshold of 1, the update for that step is skipped. This adaptive strategy stabilizes early training while maintaining convergence efficiency later on.

B.3 Baseline Details

WAN-2.1 Causal-FT is the autoregressive variant of WAN (Wan et al., 2025). Specifically, we adopt Wan2.1-T2V-1.3B, a Flow Matching–based model, as the base model. Following the CausVid initialization strategy (Yin et al., 2025), the base model is fine-tuned with causal attention masking on 16k ODE solution pairs generated from the model itself. In practice, we leverage the ODE initialization checkpoint released with the official Self-Forcing (Huang et al., 2025) repository, which corresponds exactly to the configuration of our WAN-2.1 Causal-FT setup.

NOVA AR (Deng et al., 2024) is an autoregressive video generator that does not rely on vector quantization. It reformulates video generation as non-quantized autoregressive modeling that performs temporal frame-by-frame prediction while generating spatial token sets within each frame in a flexible, set-by-set manner. To support autoregressive modeling with continuous tokens, NOVA leverages a lightweight diffusion head that models the distribution of each continuous token (Li et al., 2024). In this work, we directly compare the pure AR version of NOVA, where the model predicts each latent frame with diffusion for a fair comparison.

Model	Total	Quality	Semantic	Aesthetic	Object	Human	Spatial	Scene
Autoregressive (Diffusion) models								
NOVA AR† (Deng et al., 2024)	75.31	77.46	66.70	56.04	79.68	94.20	66.07	47.83
WAN 2.1-Causal FT†	74.96	77.41	65.15	56.04	76.51	94.20	53.25	47.83
Normalizing Flows								
STARFlow-V† (Ours)	**79.70**	**80.76**	**75.43**	**59.73**	**80.61**	**98.13**	**76.08**	**48.21**

Table 3 **Performance comparison of autoregressive video generation models on VBench (Huang et al., 2024).** Following Yang et al. (2025), we evaluate with the official GPT-augmented prompts (noted as †)

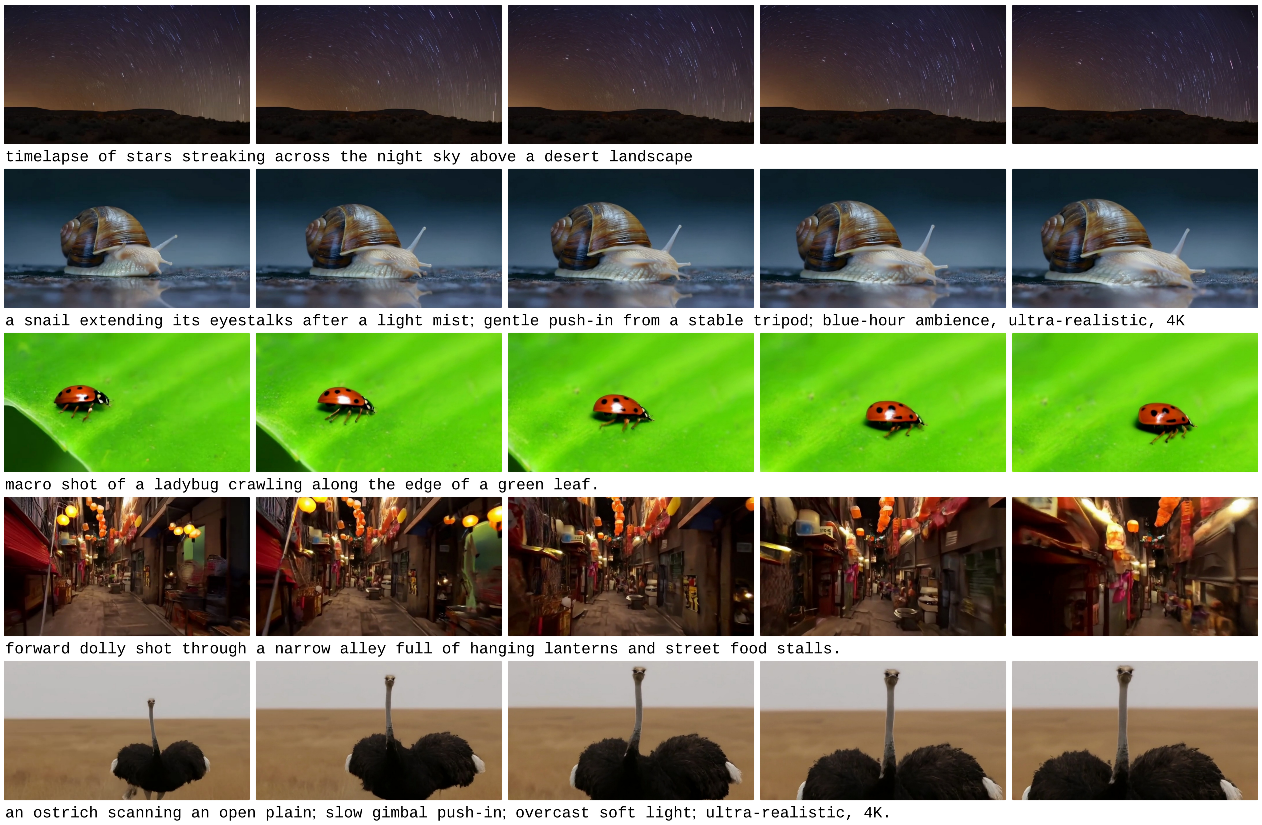


Figure 7 Generated samples from STARFlow-V given text prompts. All videos are at 480p 16fps and 5s.

C Additional Experimental Details and Results

C.1 Quantitative Comparison with Autoregressive Diffusion baselines

To evaluate the robustness of video generation under autoregressive generation, we compare STARFlow-V with autoregressive diffusion models, including NOVA AR (Deng et al., 2024) and WAN 2.1-Causal FT. Here, NOVA AR refers to the fully autoregressive video generation variant which is different from the reported in the official paper. Table 3 compares these models across a diverse set of evaluation dimensions defined in VBench (Huang et al., 2024). As shown in Table 3, STARFlow-V substantially outperforms the autoregressive diffusion baselines across all dimensions. Both NOVA AR and WAN 2.1-Causal FT exhibit clear signs of autoregressive degradation in their generated videos. Specifically, NOVA AR suffers from pronounced error accumulation, leading to increasing blur and content collapse as the video progresses. And WAN 2.1-Causal FT produces noticeable temporal inconsistency and flickering throughout the video. These failure modes are reflected in their lower scores, underscoring the difficulty of maintaining robustness in autoregressive video generation. And it further highlights the strength of our approach.

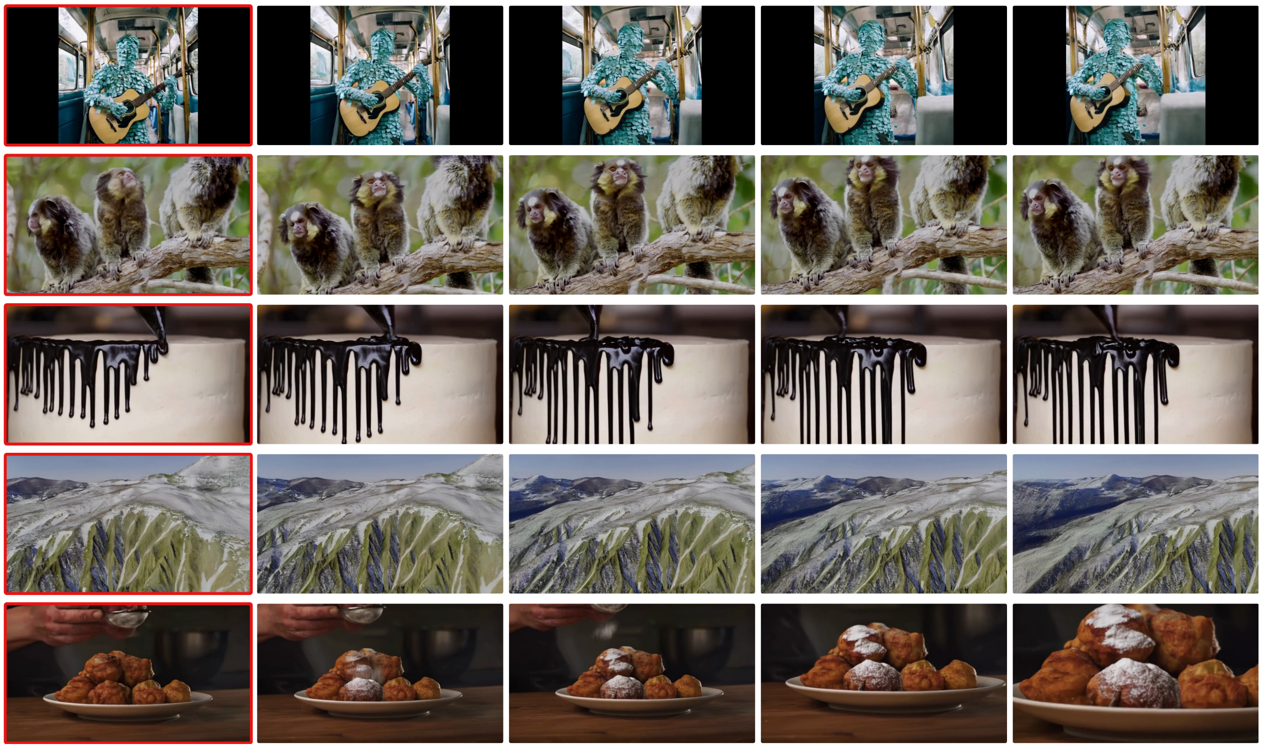


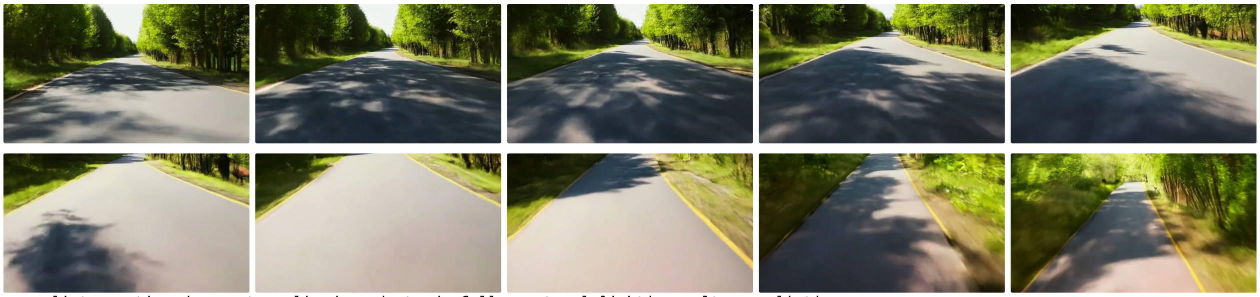
Figure 8 Generated samples from STARFlow-V given the first frame. All videos are at 480p 16fps and 5s.

C.2 Video-to-Video Generation

To support video-to-video generation and editing, we additionally finetune the pretrained STARFlow-V (7B, 384P, 81 frames) on the Señorita (Zi et al., 2025), a large-scale and high-quality instruction-based video editing dataset spanning 18 well-defined editing subcategories. Each training sample in Señorita consists of a 33-frame input video paired with a 33-frame edited target video. The model is also trained on videos with 16fps. This finetuning stage equips STARFlow-V with precise editing capabilities while preserving temporal coherence and motion consistency. During finetuning, we concatenate the input and target videos along the temporal dimension to form a single training sequence.

C.3 Additional Samples

We show additional samples at Figures 7 to 9. Besides, we provide more video generation comparison in our official codebase at https://github.com/apple/ml-starflow.



`a cyclist coasting down a tree-lined road steady follow natural lighting, ultra-realistic`
`16fps - 10s`

`school of koi swirling just below pond surface; top-down gimbal drift; occasional surface glare flare, ripples distorting bodies`
`16fps - 10s`

`a barn owl gliding through a pine forest; locked-off frame where the world moves through; micro water droplets near the lens; blue-hour cool ambience, ultra-realistic, 4K resolution; tiny particles suspended in the air.`
`16fps - 30s`

Figure 9 Generated samples from STARFlow-V given text prompts and extended with overlapping frames. For each segment, we generate 21 latent frames with 4 latent frames in overlap. Both videos are at 480p 16fps.