

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

Supplementary Material

A. Derivations and Algorithms

A.1. Derivation of STARFlow-V.

(1) Why an autoregressive Gaussian model in u is a normalizing flow. Let $T_\theta : u \mapsto z$ 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,

$$z_i = \frac{u_i - \mu_\theta(u_{<i})}{\sigma_\theta(u_{<i})}, \quad \sigma_\theta(\cdot) > 0, \quad (\text{A.1})$$

with inverse

$$u_i = \sigma_\theta(u_{<i}) z_i + \mu_\theta(u_{<i}). \quad (\text{A.2})$$

Because each z_i depends only on (u_1, \dots, u_i) and $\sigma_\theta > 0$, T_θ is bijective and continuously differentiable. The Jacobian is lower triangular with diagonal entries $\partial z_i / \partial u_i = 1 / \sigma_\theta(u_{<i})$, thus

$$\log |\det J_{T_\theta}(u)| = - \sum_i \log \sigma_\theta(u_{<i}). \quad (\text{A.3})$$

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

$$\begin{aligned} \log p_D(u) &= \log p_0(T_\theta(u)) + \log |\det J_{T_\theta}(u)| \\ &= -\frac{1}{2} \sum_i z_i^2 - \sum_i \log \sigma_\theta(u_{<i}) + \text{const}, \end{aligned} \quad (\text{A.4})$$

which is essentially the regression objective through maximum likelihood estimation over 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)),

$$\begin{aligned} p_\theta(x) &= \prod_{n=1}^N p_D(u_n | u_{<n}) |\det J_{f_S}(x_n)| \\ u_n &= f_S(x_n). \end{aligned} \quad (\text{A.5})$$

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

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

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

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

(3) Noise & denoising: what the model looks like. Following the noise-augmented training (§3.2), let $\tilde{x} = x + \sigma \epsilon$, $\epsilon \sim \mathcal{N}(0, I)$. The Tweedie single-step denoiser in the flow setting (Equation (3.3)) suggests the update $x \approx \tilde{x} + \sigma^2 \nabla_{\tilde{x}} \log p_\theta(\tilde{x})$. 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

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

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

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 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.

Algorithm 2 Autoregressive sampling ($z \rightarrow u \rightarrow x$)

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

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

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

Require: base latent z ; initial guess $u^{(0)}$ (e.g., zeros or teacher-forced prefix); block partition $\mathcal{B} = \{B_1, \dots, B_J\}$ (non-overlapping, block-causal); max iters T ; tolerance τ

- 1: **for** $t = 0, 1, 2, \dots, T - 1$ **do**
- 2: **for all** blocks $B \in \mathcal{B}$ **in parallel do**
- 3: **for all** $i \in B$ **in parallel do**
- 4: $(\mu_i^{(t)}, \sigma_i^{(t)}) \leftarrow f_D(u_{\prec i}^{(t)})$
- 5: $u_i^{(t+1)} \leftarrow \sigma_i^{(t)} z_i + \mu_i^{(t)}$
- 6: **end for**
- 7: **end for**
- 8: $r \leftarrow \frac{\|u^{(t+1)} - u^{(t)}\|_2}{\|u^{(t)}\|_2 + \varepsilon}$
- 9: **if** $r \leq \tau$ **then break**
- 10: **end if**
- 11: **end for**
- 12: **Shallow inverse:** $x \leftarrow f_S^{-1}(u^{(t+1)})$
- 13: **(One-step corrector)** $x \leftarrow x + \sigma_{\text{test}} s_\phi(x)$
- 14: **return** x

	3B	7B
Params	$\sim 3B$	$\sim 7B$
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.

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 $p_0(z)$

- 1: Initialize caches $KV \leftarrow \emptyset$, latent buffer $U \leftarrow \emptyset$
- 2: **for** $t = 1$ to T **do**
- 3: **Sample base:** $z_t \sim \mathcal{N}(0, I)$ for the next frame (or token block)
- 4: **Deep inverse:** using cached state, compute $u_t \leftarrow f_D^{-1}(z_t; KV)$ and update the KV cache.
- 5: **Shallow inverse:** $x_t \leftarrow f_S^{-1}(u_t)$
- 6: **Emit** x_t
- 7: **Maintain sliding window:** push \hat{u}_t into buffer U ; if $|U| > W$ pop the oldest and refresh the KV by re-encoding u .
- 8: **end for**
- 9: **return** $\{x_t\}_{t=1}^T$

B. Implementation Details

B.1. Architecture Design

3B. Same size as STARFlow but for *video*. The deep block f_D uses width 3072 (depth L_D , heads H_D). The 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	Total	Quality	Semantic	Aesthetic	Object	Human	Spatial	Scene
<i>Autoregressive (Diffusion) models</i>								
NOVA AR [†] [10]	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
<i>Normalizing Flows</i>								
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 [30].** Following Yang et al. [65], we evaluate with the official GPT-augmented prompts (noted as [†])

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 [18], 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.
- **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. [18], 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 [58]. Specifically, we adopt Wan2.1-T2V-1.3B, a Flow Matching-based model, as the base model. Following the CausVid initialization strategy [67], 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 [28] repository, which corresponds exactly to the configuration of our WAN-2.1 Causal-FT setup.

NOVA AR [10] 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 [37]. In this work, we directly compare the pure AR version of NOVA, where the model predicts each latent frame with diffusion.

C. Additional Results

C.1. Quantitative Comparison with AR baselines

To evaluate the robustness of video generation under autoregressive generation, we compare STARFlow-V with autoregressive diffusion models, including NOVA AR [10] and WAN 2.1-Causal FT. Here, NOVA AR refers to the fully autoregressive video generation variant. Table 3 compares these models across a diverse set of evaluation dimensions defined in VBench [30]. 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.

C.2. Inference-Time Comparison

We compare end-to-end inference efficiency on a single NVIDIA H200 GPU with batch size 1 in Tab. 4. All measurements are taken on the same hardware setup and include the complete generation pipeline. We report latency in seconds per generated video and throughput in frames per second (FPS).

Model	#Params	Resolution (H×W×F)	Throughput (FPS) ↑	Latency (s) ↓
CogVideoX	5B	720×480×49	0.45	109.208
Wan2.1	1.3B	832×480×81	0.82	99.159
STARFlow-V	7B	848×480×81	0.33	243.369

Table 4. End-to-end video generation time.

C.3. 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 [75], 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.

For additional qualitative results and video demonstrations, please refer to the HTML viewer included in the supplementary materials.

C.4. Qualitative Results

Please check the supplemental materials (via [index.html](#)) for more video results on various tasks, comparison with baselines and current limitations.