STREAM-DIFFVSR: LOW-LATENCY STREAMABLE VIDEO SUPER-RESOLUTION VIA AUTO-REGRESSIVE DIFFUSION

Anonymous authors
Paper under double-blind review

Figure 1: **Comparison of visual quality and inference speed across various categories of VSR methods.** Stream-DiffVSR achieves superior perceptual quality (lower LPIPS) and maintains comparable runtime to CNN- and Transformer-based online models, while also demonstrating significantly reduced inference latency compared to existing offline approaches. Best and second-best results are marked in red and green.

ABSTRACT

Diffusion-based video super-resolution (VSR) methods have recently demonstrated remarkable perceptual quality; however, their reliance on future-frame information and computationally expensive iterative denoising has restricted their application in latency-sensitive contexts. We present Stream-DiffVSR, a causally conditioned diffusion VSR framework designed for efficient online inference. Our method operates strictly with past frames and integrates three key components: a four-step distilled denoiser, an auto-regressive temporal guidance (ARTG) module that injects motion-aligned temporal cues into the denoising process, and a lightweight temporal-aware decoder with temporal processor module (TPM) that enhances spatial detail and temporal consistency. Stream-DiffVSR processes 720p frames in just 0.328 seconds on an RTX 4090 GPU, significantly outperforming previous diffusion-based methods. Compared with state-of-the-art online methods such as TMP (Zhang et al., 2024b), Stream-DiffVSR achieves a substantial improvement in perceptual quality (LPIPS improved by 0.095) while reducing inference latency by more than 130X relative to previous diffusion-based VSR approaches. These results demonstrate the potential of diffusion models for practical deployment in time-sensitive rendering pipelines and real world video super-resolution systems. Notably, Stream-DiffVSR achieves the lowest latency ever reported among diffusion-based VSR methods, reducing the initial delay from over 4600 seconds to just 0.328 seconds. This makes it the first diffusion-based solution viable for real-time online deployment.

1 INTRODUCTION

Video super-resolution (VSR) aims to reconstruct high resolution (HR) videos from low resolution (LR) inputs and plays a critical role in applications such as surveillance, live broadcasting, video conferencing, autonomous driving, drone imaging, and increasingly, low latency rendering workflows, such as neural rendering and resolution upscaling in game engines and AR/VR systems where latency-aware processing is crucial for visual continuity.

Table 1: **Comparison of diffusion-based VSR methods.** We report online capability, inference steps, runtime (FPS on 720p, RTX 4090), maximum end-to-end latency (sec), and whether each method uses distillation, temporal modeling, or offline future frames. OOM denotes out-of-memory, and **-** indicates missing public inference results. Notably, Stream-DiffVSR is the only diffusion-based method that runs in a strictly online, past-only setting with the lowest latency.

Method	Online	# of Steps	FPS @720p	Max latency	Distill	Temporal Input	Temporal Decoder
StableVSR (Rota et al., 2024)	x	50	0.02	4620	x	Future/Bi-dir	x
MGLD-VSR (Yang et al., 2024)	x	50	0.02	218	x	Future/Bi-dir	✓
Upscale-A-Video (Zhou et al., 2024a)	x	30	OOM	-	x	Future/Bi-dir	✓
DiffVSR (Li et al., 2025)	x	–	–	-	x	Future/Bi-dir	✓
VEnhancer (He et al., 2024)	x	15	OOM	-	x	Future/Bi-dir	✓
Stream-DiffVSR (ours)	✓	**4**	**3.05**	0.328	✓	**Past-only**	✓

Specifically, latency-sensitive processing involves two key aspects: per-frame inference time (throughput) and end-to-end system latency (delay between receiving an input frame and producing its output). Existing VSR methods often struggle with this trade-off. While CNN- and Transformer-based models offer a balance between efficiency and quality, they fall short in perceptual detail. Diffusion-based models excel in perceptual quality due to strong generative priors, but suffer from high computational cost and reliance on future frames, making them impractical for time-sensitive video applications.

In this paper, we propose **Stream-DiffVSR**, a diffusion-based method specifically tailored to online video super-resolution, effectively bridging the gap between high-quality but slow diffusion methods and fast but lower quality CNN- or Transformer-based methods. Unlike previous diffusion-based VSR approaches (e.g., StableVSR (Rota et al., 2024) and MGLD-VSR (Yang et al., 2024)) that typically require 50 or more denoising steps and bidirectional temporal information, our method leverages diffusion model distillation to significantly accelerate inference by reducing denoising steps to just four. Additionally, we introduce an Auto-regressive Temporal Guidance mechanism and an Auto-regressive Temporal-aware Decoder to effectively exploit temporal information from previous frames, significantly enhancing temporal consistency and perceptual fidelity.

Fig. 1 illustrates the core advantage of our approach by comparing visual quality and runtime across various categories of video super-resolution methods. Our Stream-DiffVSR achieves superior perceptual quality (measured by LPIPS (Zhang et al., 2018)) and temporal consistency, outperforming existing unidirectional CNN- and Transformer-based methods (e.g., MIA-VSR (Zhou et al., 2024b), RealViformer (Zhang & Yao, 2024), TMP (Zhang et al., 2024b)). Notably, Stream-DiffVSR offers significantly faster per-frame inference than prior diffusion-based approaches (e.g., StableVSR (Rota et al., 2024), MGLD-VSR (Yang et al., 2024)), attributed to our use of a distilled 4-step denoising process and a lightweight temporal-aware decoder.

In addition, existing diffusion-based methods, such as StableVSR (Rota et al., 2024) typically rely on bidirectional or future-frame information, resulting in prohibitively high processing latency that is not suitable for online scenarios. Specifically, for a 100-frame video, StableVSR (46.2 s/frame) would incur an initial latency exceeding 4600 seconds on an RTX 4090 GPU, as it requires processing the entire sequence before generating even the first output frame. In contrast, our Stream-DiffVSR operates in a strictly causal, autoregressive manner, conditioning only on the immediately preceding frame. Consequently, the initial frame latency of Stream-DiffVSR corresponds to a single frame's inference time (0.328 s/frame), reducing the latency by more than three orders of magnitude compared to StableVSR. This significant latency reduction demonstrates that Stream-DiffVSR effectively unlocks the potential of diffusion models for practical, low-latency online video super-resolution.

To summarize, the main contributions of this paper are:

- We introduce the first diffusion-based framework explicitly designed for online, low-latency video super-resolution, achieving efficient inference through distillation from 50 denoising steps down to 4 steps.
- We propose a novel Auto-regressive Temporal Guidance mechanism and a Temporal-aware Decoder to effectively leverage temporal information *only* from past frames, significantly enhancing perceptual quality and temporal consistency.
- Extensive experiments demonstrate that our approach outperforms existing methods across key perceptual and temporal consistency metrics while achieving practical inference speeds, thereby making diffusion-based VSR applicable for real-world online scenarios.

To contextualize our contributions, Table 1 compares recent diffusion-based VSR methods in terms of online inference capability, runtime efficiency, and temporal modeling. Our method uniquely achieves online low-latency inference while preserving high visual quality and temporal stability. This substantial latency reduction of over three orders of magnitude compared to prior diffusion-based VSR models demonstrates that Stream-DiffVSR is uniquely suited for low-latency online applications such as video conferencing and AR/VR.

2 Related Work

Video Super-resolution. VSR methods reconstruct high-resolution videos from low-resolution inputs through CNN-based approaches (Xue et al., 2019; Tian et al., 2020; Wang et al., 2019; Chan et al., 2021; 2022a), deformable convolutions (Tian et al., 2020; Dai et al., 2017; Zhu et al., 2019), online processing (Zhang et al., 2024b), recurrent architectures (Sajjadi et al., 2018; Fuoli et al., 2019; Isobe et al., 2020; Yi et al., 2019; Li et al., 2020), flow-guided methods (Youk et al., 2024; Guo et al., 2024), and Transformer-based models (Vaswani et al., 2017; Liang et al., 2022b;a; Shi et al., 2022; Zhou et al., 2024b). Despite advances, low-latency online processing remains challenging.

Real-world Video Super-resolution. Real-world VSR addresses unknown degradations (Yang et al., 2021; Chan et al., 2022b) through pre-cleaning modules (Chan et al., 2022b; Goodfellow et al., 2020; Wang et al., 2021), online approaches (Zhang & Yao, 2024), kernel estimation (Pan et al., 2021; Ji et al., 2020), synthetic degradations (Jeelani et al., 2023; Song et al., 2024; Zhang et al., 2023), new benchmarks (Zhao et al., 2025; Conde et al., 2024), real-time systems (Cao et al., 2021), advanced GANs (Chen et al., 2024), and Transformer restorers (Zamir et al., 2022; Liang et al., 2021; Blau & Michaeli, 2018). Warp error-aware consistency (Lei et al., 2020) emphasizes temporal error regularisation.

Diffusion-based Image and Video Restoration. Diffusion models provide powerful generative priors (Rombach et al., 2021; Esser et al., 2021) for single-image SR (Saharia et al., 2022; Li et al., 2022), inpainting (Lugmayr et al., 2022; Weng et al., 2024), and quality enhancement (Ho et al., 2022; Gao et al., 2023; Wang et al., 2024c). Video diffusion methods include StableVSR (Rota et al., 2024), MGLD-VSR (Yang et al., 2024), DC-VSR (Han et al., 2025), DOVE (Chen et al., 2025), UltraVSR (Liu et al., 2025), Upscale-A-Video (Zhou et al., 2024a), DiffVSR (Li et al., 2025), VideoGigaGAN (Xu et al., 2024), VEnhancer (He et al., 2024), temporal coherence (Wang et al., 2025), and AVID (Zhang et al., 2024c). Auto-regressive approaches (Sun et al., 2025b; Xie et al., 2025; Liu et al., 2024) show promise. Acceleration techniques include consistency models (Luo et al., 2023; Geng et al., 2024), advanced solvers (Lu et al., 2022; Lu et al., 2025; Zheng et al., 2023), flow-based methods (Liu et al., 2023; Jin et al., 2024), distillation (Salimans & Ho, 2022; Meng et al., 2023; Zhou et al., 2024c; Xie et al., 2024), and theoretical advances (Wang et al., 2024a;b). Recent image/offline distillation methods (Sun et al., 2025a; Zhang et al., 2024a; Wu et al., 2024a;b) exist, but our *Stream-DiffVSR* uniquely applies distillation in strict online settings with causal temporal modeling for real-time VSR.

3 Method

We propose Stream-DiffVSR, a streamable auto-regressive diffusion framework for efficient video super-resolution (VSR). The key innovation is its auto-regressive design, which explicitly enhances temporal consistency and inference speed. Our method comprises: (1) a distilled few-step U-Net for accelerated diffusion inference, (2) Auto-regressive Temporal Guidance that conditions latent denoising on previously warped high-quality frames, and (3) an Auto-regressive Temporal-aware Decoder explicitly incorporating temporal information. Together, these components enable Stream-DiffVSR to generate stable and perceptually coherent videos.

3.1 Diffusion Models Preliminaries

Diffusion Models (Ho et al., 2020) transform complex data distributions into simpler Gaussian distributions via a forward diffusion process and reconstruct the original data using a learned reverse denoising process. The forward process gradually adds Gaussian noise to the initial data x_0, forming a Markov chain: $q(x_t \mid x_{t-1}) = \mathcal{N}\big(x_t; \sqrt{1-\beta_t}\, x_{t-1},\, \beta_t I\big)$ for $t = 1, \ldots, T$, where β_t denotes a predefined noise schedule. At timestep t, the noised data x_t can be directly sampled from the clean data x_0 as: $x_t = \sqrt{\alpha_t}\, x_0 + \sqrt{1-\alpha_t}\, \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$ and $\alpha_t = \prod_{i=1}^{t}(1-\beta_i)$, where $\alpha_t = \prod_{i=1}^{t}(1-\beta_i)$. The reverse process progressively removes noise from x_T, recon-

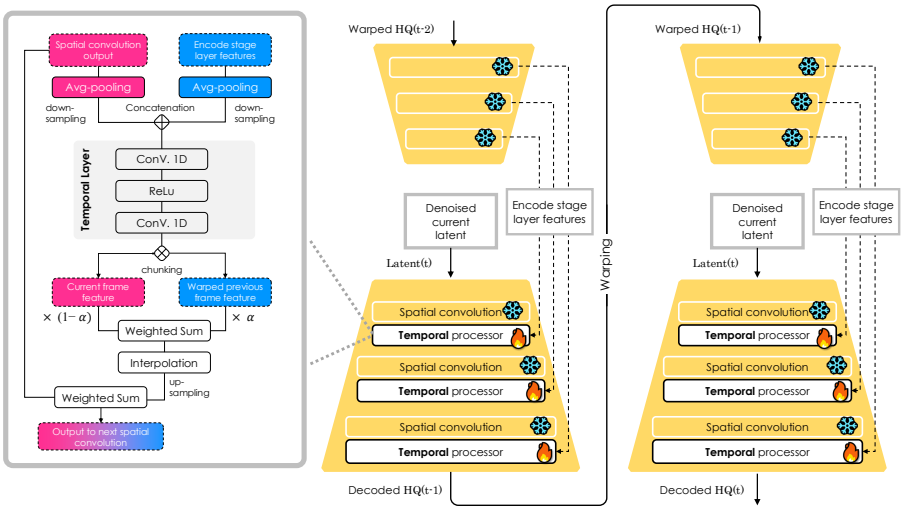

Figure 2: **Overview of Auto-regressive Temporal-aware Decoder.** Given the denoised latent and warped previous frame, our decoder enhances temporal consistency using temporal processor modules. These module aligns and fuses these features via interpolation, convolution, and weighted fusion, effectively stabilizing detail reconstruction when decoding into the final RGB frame.

structing the original data x_0 through a learned denoising operation modeled as a Markov chain, i.e., $p_\theta(x_0, \ldots, x_{T-1} \mid x_T) = \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t)$. Each individual step is parameterized by a neural network-based denoising function $p_\theta(x_{t-1} \mid x_t) = \mathcal{N}\big(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(t)I\big)$. Typically, the network predicts the noise component $\epsilon_\theta(x_t, t)$, from which the denoising mean is estimated as $\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\Big(x_t - \frac{1-\alpha_t}{\sqrt{1-\alpha_t}}\,\epsilon_\theta(x_t, t)\Big)$. Latent Diffusion Models (LDMs) (Rombach et al., 2022b) further reduce computational complexity by projecting data into a lower-dimensional latent space using Variational Autoencoders (VAEs), significantly accelerating inference without sacrificing generative quality.

3.2 U-Net Rollout Distillation

We distill a pre-trained Stable Diffusion (SD) x4 Upscaler (Rombach et al., 2022b;a), originally designed for 50-step inference, into a 4-step variant balancing speed and perceptual quality. To close the training–inference gap of timestep-sampling distillation, we adopt rollout distillation, where the U-Net performs the full 4-step denoising each iteration until a clean latent is obtained. Detailed algorithms and implementation are provided in the supplementary material due to page constraints.

Unlike conventional distillation that supervises random intermediate timesteps, our method applies loss only on the final denoised latent, ensuring the training trajectory mirrors inference and improving stability and alignment.

Our distillation requires no architectural changes. We train the U-Net by optimizing latent reconstruction with a loss that balances spatial accuracy, perceptual fidelity, and realism:

$$\mathcal{L}_{\text{distill}} = \|\mathbf{z}_{\text{den}} - \mathbf{z}_{\text{gt}}\|_2^2 + \lambda_{\text{LPIPS}} \cdot \text{LPIPS}\left(D(\mathbf{z}_{\text{den}}), \mathbf{x}_{\text{gt}}\right) + \lambda_{\text{GAN}} \cdot \mathcal{L}_{\text{GAN}}\left(D(\mathbf{z}_{\text{den}})\right), \tag{1}$$

where $\mathbf{z}_{\text{den}}$ and $\mathbf{z}_{\text{gt}}$ are the denoised and ground-truth latent representations. The decoder $D(\cdot)$ maps latent features back to RGB space for perceptual (LPIPS) and adversarial (GAN) loss calculations, encouraging visually realistic outputs.

3.3 Auto-regressive Temporal Guidance

Leveraging temporal information is crucial for capturing dynamics and ensuring frame continuity in video super-resolution. However, extensive use of temporal cues often introduces substantial computational overhead, resulting in increased per-frame inference time and system latency. Therefore, designing efficient online VSR systems requires a careful balance between temporal information utilization and computational efficiency to support low-latency processing.

To this end, we propose **Auto-regressive Temporal Guidance (ARTG)**, which enforces temporal coherence during latent denoising. At each timestep t, the U-Net takes both the current noised latent

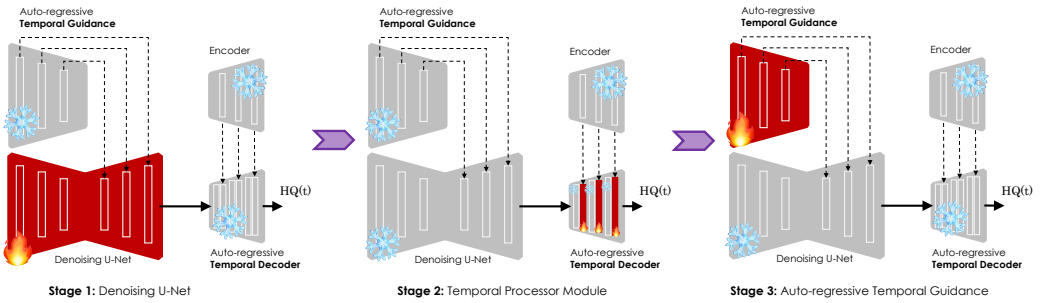

Figure 3: **Training pipeline of Stream-DiffVSR.** The training process consists of three sequential stages: (1) Distilling the denoising U-Net to reduce diffusion steps while maintaining perceptual quality with training objective (1); (2) Training the Temporal Processor Module (TPM) within the decoder to enhance temporal consistency at the RGB level with training objective (3); (3) Training the Auto-Regressive Temporal Guidance (ARTG) module to leverage previously restored high-quality frames for improved temporal coherence with training objective (6). Each module is trained separately before integrating them into the final framework.

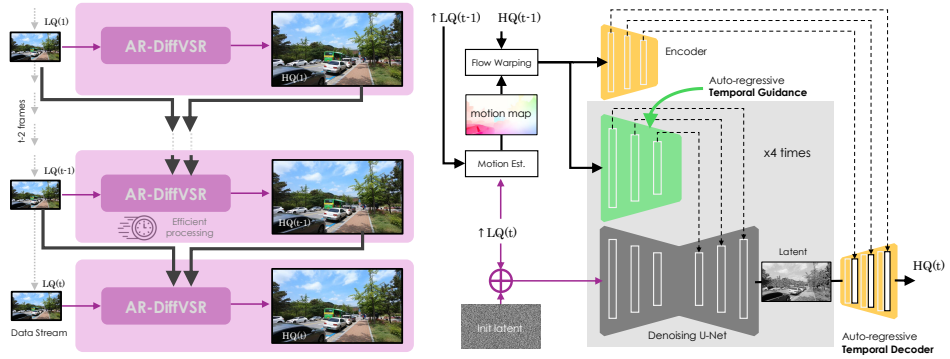

Figure 4: **Overview of our pipeline.** Given a low-quality (LQ) input frame, we first initialize its latent representation and employ an autoregressive diffusion model composed of a distilled denoising U-Net, autoregressive temporal Guidance, and an autoregressive temporal Decoder. Temporal guidance utilizes flow-warped high-quality (HQ) results from the previous frame to condition the current frame's latent denoising and decoding processes, significantly improving perceptual quality and temporal consistency in an efficient, online manner.

z_t and the warped RGB frame from the previous output, $\hat{x}_{t-1}^{\text{warp}} = \text{Warp}(x_{t-1}^{\text{SR}}, f_{t\leftarrow t-1})$, where $f_{t\leftarrow t-1}$ is the optical flow from frame $t-1$ to t. The denoising prediction is then formulated as:

$$\hat{\epsilon}_\theta = \text{UNet}(z_t, t, \hat{x}_{t-1}^{\text{warp}}), \tag{2}$$

where the warped image $\hat{x}_{t-1}^{\text{warp}}$ serves as temporal conditioning input to guide the denoising process.

We train the **ARTG** module independently using consecutive pairs of low-quality and high-quality frames. The denoising U-Net and decoder are kept fixed during this stage, and the training objective focuses on reconstructing the target latent representation while preserving perceptual quality and visual realism. The total loss function is defined as:

$$\mathcal{L}_{\text{ARTG}} = \|\mathbf{z}_{\text{den}} - \mathbf{z}_{\text{gt}}\|_2^2 + \lambda_{\text{LPIPS}} \cdot \text{LPIPS}(D(\mathbf{z}_{\text{den}}), \mathbf{x}_{\text{gt}}) + \lambda_{\text{GAN}} \cdot \mathcal{L}_{\text{GAN}}(D(\mathbf{z}_{\text{den}})), \tag{3}$$

where $\mathbf{z}_{\text{den}}$ denotes the denoised latent from DDIM updates with predicted noise $\hat{\epsilon}_\theta$, and $\mathbf{z}_{\text{gt}}$ is the ground-truth latent. The decoder $D(\cdot)$ maps latents to RGB, yielding $D(\mathbf{z}_{\text{den}})$ for comparison with the ground-truth image $\mathbf{x}_{\text{gt}}$. The latent ℓ_2 loss enforces pixel-wise alignment, the perceptual loss ensures visual fidelity, and the adversarial loss promotes realism. This design leverages only past frames to propagate temporal context, improving consistency without extra latency.

3.4 Auto-regressive Temporal-aware Decoder

Although the **Auto-regressive Temporal Guidance (ARTG)** enhances temporal consistency in the latent space, the features generated by the Stable Diffusion $\times 4$ Upscaler reside at one-quarter of the target resolution. This resolution gap may lead to decoding artifacts or misalignment in dynamic scenes.

Table 2: **Quantitative comparison against bidirectional/offline methods on the REDS4 dataset.** We compare CNN-, Transformer-, and diffusion-based methods on REDS4. Stream-DiffVSR achieves superior perceptual and temporal quality with high stability across sequences. ↑ indicates higher is better; ↓ indicates lower is better. **Dir.** denotes temporal direction: **B** for bidirectional/offline, **U** for unidirectional/online. Runtime is measured per 720p frame on an RTX 4090. **Latency-max** denotes the maximum end-to-end latency measured over 100-frame video sequences, providing a fair comparison with offline methods whose initial delay scales with sequence length. **tLP** and **tOF** are scaled by 100× and 10×. Best and second-best results are marked in red and blue.

Dir.	Method	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	MUSIQ↑	NIQE↓	NRQM↑	BRISQUE↓	tLP↓	tOF↓	Runtime (s)↓	latency-max (s)↓
CNN-based Methods													
-	Bicubic	25.501	0.712	0.460	0.187	27.362	7.360	3.459	60.256	21.603	4.241	-	-
B	BasicVSR++	32.386	0.907	0.132	0.069	67.002	3.850	6.363	38.641	9.017	2.490	0.098	9.8
B	RealBasicVSR	27.042	0.778	0.134	0.065	67.033	2.530	6.769	18.046	6.422	4.759	0.064	6.4
Transformer-based Methods													
B	RVRT	32.701	0.911	0.130	0.067	67.251	3.793	6.366	38.038	9.133	2.421	0.498	2.49
B	MIA-VSR	32.790	0.912	0.123	0.064	68.140	3.742	6.451	37.099	8.870	2.354	0.768	76.8
Diffusion-based Methods													
B	StableVSR	27.928	0.793	0.102	0.047	67.058	2.713	6.960	16.249	5.755	2.742	46.2	4620
B	MGLD-VSR	26.53	0.749	0.151	0.065	66.081	2.972	6.701	15.291	18.139	5.910	43.6	218
U	Ours	27.256	0.766	0.099	0.062	65.595	3.114	7.055	17.717	4.198	3.638	0.328	0.328

Table 3: **Quantitative comparison against unidirectional/online methods on the REDS4 dataset.**

Dir.	Method	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	MUSIQ↑	NIQE↓	NRQM↑	BRISQUE↓	tLP↓	tOF↓	Runtime (s)↓	latency-max (s)↓
CNN-based Methods													
-	Bicubic	25.501	0.712	0.460	0.187	27.362	7.360	3.459	60.256	21.603	4.241	-	-
U	TMP	30.672	0.871	0.194	0.090	63.818	4.378	5.796	43.394	10.424	2.480	0.041	0.041
Transformer-based Methods													
U	RealViformer	26.763	0.761	0.129	0.065	64.585	2.731	7.028	17.272	11.261	4.037	0.099	9.9
Diffusion-based Methods													
U	StableVSR*	27.174	0.763	0.111	0.051	66.428	2.572	6.944	15.805	11.107	3.925	46.2	4620
U	Ours	27.256	0.766	0.099	0.062	65.595	3.114	7.055	17.117	4.198	3.638	0.328	0.328

To address this issue, we propose an **Auto-regressive Temporal-aware Decoder** that incorporates temporal context into decoding to enhance spatial fidelity and temporal consistency. At timestep t, the decoder takes the denoised latent $\mathbf{z}_t^{\text{den}}$ and the aligned feature $\hat{\mathbf{f}}_{t-1}$ derived from the previous super-resolved frame. Specifically, we compute:

$$\hat{\mathbf{x}}_{t-1}^{\text{warp}} = \text{Warp}(\mathbf{x}_{t-1}^{\text{SR}}, f_{t\leftarrow t-1}), \quad \hat{\mathbf{f}}_{t-1} = \text{Enc}(\hat{\mathbf{x}}_{t-1}^{\text{warp}}), \tag{4}$$

where $\mathbf{x}_{t-1}^{\text{SR}}$ is the previously generated RGB output, $f_{t\leftarrow t-1}$ is the optical flow from frame $t-1$ to t, and $\text{Enc}(\cdot)$ is a frozen encoder that projects the warped image into the latent feature space.

The decoder then synthesizes the current frame using:

$$\mathbf{x}_t^{\text{SR}} = \text{Decoder}(\mathbf{z}_t^{\text{den}}, \hat{\mathbf{f}}_{t-1}). \tag{5}$$

We adopt a multi-scale fusion strategy inside the decoder to combine current spatial information and prior temporal features across multiple resolution levels, as illustrated in Fig. 2. This design helps reinforce temporal coherence while recovering fine spatial details.

Temporal Processor Module (TPM). We integrate TPM after each spatial convolutional layer in the decoder to explicitly inject temporal coherence, enhancing stability and continuity of reconstructed frames. These modules utilize latent features from the current frame and warped features from the previous frame, optimizing temporal consistency independently from spatial reconstruction. Our training objective for the TPM is defined as:

$$\mathcal{L}_{\text{TPM}} = \mathcal{L}_{\text{rec}}(\mathbf{x}_t^{\text{rec}}, \mathbf{x}_t^{\text{GT}}) + \lambda_{\text{flow}} \left\| \text{OF}(\mathbf{x}_t^{\text{rec}}, \mathbf{x}_{t-1}^{\text{rec}}) - \text{OF}(\mathbf{x}_t^{\text{GT}}, \mathbf{x}_{t-1}^{\text{GT}}) \right\|_2^2 + \lambda_{\text{GAN}} \mathcal{L}_{\text{GAN}}(\mathbf{x}_t^{\text{rec}}) + \lambda_{\text{LPIPS}} \text{LPIPS}(\mathbf{x}_t^{\text{rec}}, \mathbf{x}_t^{\text{GT}}) \tag{6}$$

where $\mathbf{x}_t^{\text{SR}} \in \mathbb{R}^{3\times H\times W}$ is the predicted frame at time t, and $\mathbf{x}_t^{\text{GT}}$ is the ground-truth frame. The reconstruction loss $\mathcal{L}_{\text{rec}} = \text{SmoothL1}(\mathbf{x}_t^{\text{rec}}, \mathbf{x}_t^{\text{GT}})$ enforces spatial fidelity, the adversarial loss $\mathcal{L}_{\text{GAN}}$ improves realism, and the optical-flow term $\text{OF}(\cdot,\cdot)$ reduces temporal discrepancies, yielding consistent and perceptually faithful outputs.

Table 4: **Quantitative comparison against bidirectional/offline methods on the Vimeo-90K-T dataset.** Our Stream-DiffVSR consistently outperforms other unidirectional methods in perceptual quality and temporal consistency, while also demonstrating substantially lower runtime. Runtime denotes the average per-frame inference time (in seconds) on 448×256 resolution videos using a single NVIDIA RTX 4090 GPU. Best and second-best results are highlighted in red and blue, respectively.

Dir.	Method	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	MUSIQ↑	NIQE↓	NRQM↑	BRISQUE↓	tLP↓	tOF↓	Runtime (s)↓	latency-max (s)↓
CNN-based Methods													
-	Bicubic	29.282	0.864	0.297	0.209	23.433	8.735	3.588	61.714	11.606	2.49	-	-
B	BasicVSR++	37.479	0.956	0.098	0.117	51.940	7.077	5.509	47.792	4.691	1.57	0.012	0.084
B	RealBasicVSR	29.388	0.857	0.156	0.149	56.986	5.069	7.413	23.822	10.947	3.46	0.008	0.056
Transformer-based Methods													
B	RVRT	37.815	0.955	0.093	0.105	49.937	7.205	5.393	48.352	4.873	1.429	0.061	0.305
B	MIA-VSR	37.598	0.957	0.086	0.101	51.402	7.116	5.569	47.865	4.696	1.419	0.096	0.672
Diffusion-based Methods													
B	StableVSR	31.823	0.878	0.095	0.111	54.582	4.745	7.265	20.039	26.224	3.108	5.749	40.243
B	MGLD-VSR	29.651	0.865	0.151	0.137	57.788	5.340	7.217	20.761	12.550	4.661	5.426	27.130
U	Ours	32.593	0.900	0.056	0.105	52.755	4.403	7.672	29.297	4.307	2.689	0.041	0.041

Table 5: **Quantitative comparison against unidirectional/online methods on the Vimeo-90K-T dataset.**

Dir.	Method	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	MUSIQ↑	NIQE↓	NRQM↑	BRISQUE↓	tLP↓	tOF↓	Runtime (s)↓	latency-max (s)↓
CNN-based Methods													
-	Bicubic	29.282	0.864	0.297	0.209	23.433	8.735	3.588	61.714	11.606	2.49	-	-
U	TMP	36.482	0.946	0.109	0.118	48.374	7.368	5.096	49.192	4.870	1.603	0.006	0.006
Transformer-based Methods													
U	RealViformer	30.291	0.877	0.130	0.140	53.107	5.515	6.711	24.628	8.232	2.769	0.013	0.091
Diffusion-based Methods													
U	StableVSR*	31.729	0.875	0.072	0.113	54.447	4.698	7.280	19.836	30.858	3.144	5.749	40.243
U	Ours	32.593	0.900	0.056	0.105	52.755	4.403	7.672	29.297	4.307	2.689	0.041	0.041

3.5 Training and Inference Stages

Our training pipeline consists of three independent stages (Fig. 3), while our inference process and the Auto-Regressive Diffusion-based VSR algorithm are illustrated in Fig. 4 and detailed in the appendix due to page constraints, respectively.

Distilling the Denoising U-Net. We first distill the denoising U-Net using pairs of low-quality (LQ) and high-quality (HQ) frames to optimize per-frame super-resolution and latent-space consistency.

Training the Temporal Processor Module (TPM). In parallel, we train the Temporal Processor Module (TPM) in the decoder using ground-truth frames, keeping all other weights fixed. This enhances the decoder's capability to incorporate temporal information into the final RGB reconstruction.

Training Auto-regressive Temporal Guidance. After training and freezing the U-Net and decoder, we train the ARTG, which leverages flow-aligned previous outputs to enhance temporal coherence without degrading spatial quality. This staged training strategy progressively refines spatial fidelity, latent consistency, and temporal smoothness in a decoupled manner.

Inference. Given a sequence of low-quality (LQ) frames, our method auto-regressively generates high-quality (HQ) outputs. For each frame t, the denoising process is conditioned on the previous frame HQ_{t-1}, warped using optical flow to capture temporal motion. To balance quality and efficiency, we adopt a **4-step DDIM denoising scheme** with a distilled U-Net. By leveraging motion alignment and fewer denoising steps, our inference framework achieves stable temporal consistency efficiently.

4 Experiment

Due to space limitations, we provide the experimental setup in the appendix.

4.1 Quantitative Evaluation

We quantitatively evaluate Stream-DiffVSR against state-of-the-art VSR methods on benchmark datasets (REDS4, Vimeo-90K-T, VideoLQ, Vid4), covering diverse content and motion complexities. Tables 2 and 4 summarize the results categorized by CNN-, Transformer-, and Diffusion-based methods, as well as bidirectional (offline) and unidirectional (online) approaches. On REDS4 (Table 2), Stream-DiffVSR achieves superior perceptual quality (LPIPS=0.099) compared to CNN (BasicVSR++, RealBasicVSR), Transformer (RVRT), and diffusion-based methods (StableVSR, MGLD-

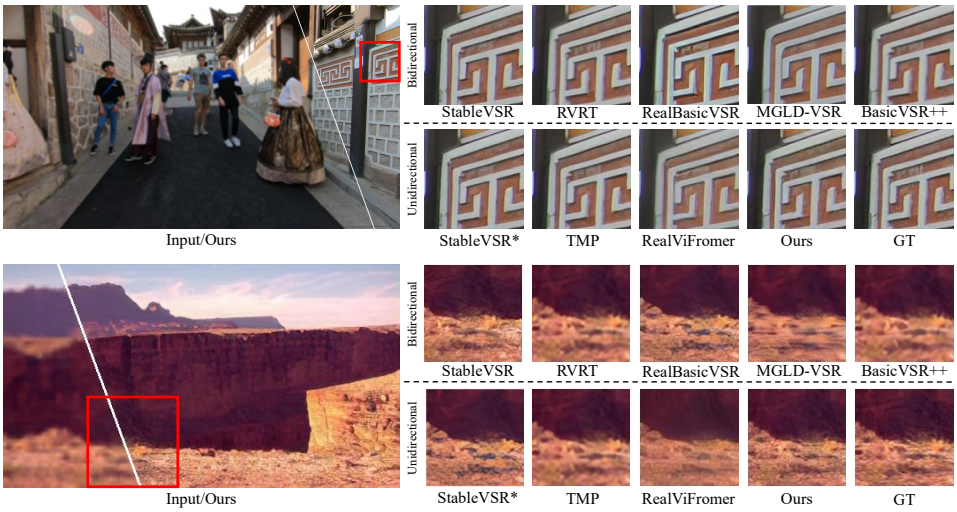

Figure 5: **Qualitative comparison on REDS4 and Vimeo-90K-T datasets.** Our method demonstrates superior visual quality with sharper details compared to unidirectional methods (TMP (Zhang et al., 2024b), RealViformer (Zhang & Yao, 2024)) and competitive performance against bidirectional methods (StableVSR (Rota et al., 2024), MGLD-VSR (Yang et al., 2024), RVRT (Liang et al., 2022b), BasicVSR++(Chan et al., 2022a), RealBasicVSR(Chan et al., 2022b)). Improvements include reduced artifacts and enhanced temporal stability (see zoomed patches).

Figure 6: **Ablation study on denoising step count within Stream-DiffVSR.** We evaluate 50, 10, 1, and 4 steps. Our 4-step design achieves a favorable balance between perceptual quality and runtime.

Denoising Step(s)	LPIPS↓	DISTS↓	MUSIQ↑	NIQE↓	NRQM↑	BRISQUE↓	tLP↓	tOF↓	Runtime (s)↓
50	0.102	0.068	66.061	2.804	7.026	9.925	18.798	3.826	3.460
10	0.122	0.072	64.900	2.869	6.917	12.461	9.990	3.625	0.718
1	0.138	0.076	63.915	3.843	6.984	29.552	9.899	3.882	0.106
4 (Ours)	0.099	0.062	65.586	3.111	7.056	17.667	4.265	3.620	0.328

Figure 7: **Ablation study on Rollout Training.** Comparison of random timestep distillation vs. our rollout strategy across fidelity (PSNR, SSIM) and perceptual metrics (LPIPS, DISTS, MUSIQ). Training time in GPU hours on 4× A6000 GPUs.

Distillation Methods	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	MUSIQ↑	GPU Hours (h)↓
Random Timestep Selection	26.27	0.743	0.099	0.071	65.981	60.5
Rollout Distillation	26.36	0.753	0.095	0.075	66.391	21

VSR). It also performs competitively on temporal consistency metrics (tLP=4.198, tOF=3.638), significantly outperforming existing unidirectional approaches. Importantly, Stream-DiffVSR attains these results with a much faster inference speed (0.328s/frame), compared to diffusion-based baselines (StableVSR: 46.2s/frame, MGLD-VSR: 43.6s/frame). Similarly, on Vimeo-90K-T (Table 5), Stream-DiffVSR excels in perceptual metrics (LPIPS=0.056, DISTS=0.105), surpassing recent unidirectional methods (MIA-VSR, RealViformer). Our method also substantially improves temporal consistency (tLP=5.307, tOF=2.689) with a competitive runtime (0.041s/frame), confirming its practicality for online applications. We additionally evaluate on VideoLQ and Vid4 to assess robustness. Stream-DiffVSR maintains strong perceptual and temporal performance across both datasets, demonstrating good generalization to real-world and classic benchmarks.

4.2 Qualitative Comparisons

We provide qualitative comparisons in Fig. 5, showing that Stream-DiffVSR produces sharper details and fewer artifacts than prior methods. Temporal consistency and flow coherence are visualized in Fig. 17 and Fig. 18, where Stream-DiffVSR yields smoother transitions and reduced flickering. We also include a qualitative comparison with Upscale-A-Video (UAV) (Zhou et al., 2024a) in appendix.

4.3 Ablation Study

To analyze the contributions of individual components in Stream-DiffVSR, we conduct ablation studies to validate the effectiveness of each component and training strategy, including denoising step reduction, ARTG, TPM, the variation of timepstep selections and stage combination in training. All ablations are performed on the REDS4 benchmark to ensure consistent evaluation across perceptual quality and temporal stability.

We perform ablation studies on training strategies in Fig. 7 and Fig. 9. For stage-wise training, partial or joint training yields inferior results, while our separate stage-wise scheme achieves the best trade-off across fidelity, perceptual, and temporal metrics. For distillation, rollout training

Figure 8: **Ablation study of temporal modules in Stream-DiffVSR.**

Component	LPIPS↓	DISTS↓	MUSIQ↑	NIQE↓	NRQM↑	BRISQUE↓	tLP↓	tOF↓	WarpErr↓
Per-frame	0.099	0.071	65.981	3.249	6.969	21.655	7.261	4.201	25.668
w/o ARTG	0.117	0.070	63.347	3.194	6.980	19.027	6.132	3.910	16.598
w/o TPM	0.116	0.078	67.110	3.197	7.007	20.279	12.847	4.639	21.990
TPM (unwarped)	0.122	0.082	63.849	3.201	7.159	14.063	12.846	5.689	17.143
Ours	0.099	0.062	65.586	3.111	7.256	17.667	4.265	3.620	14.909

Figure 10: **Ablation study on the Temporal Processor Module (TPM).** Evaluating the impact of TPM on temporal consistency. Integrating TPM effectively improves motion stability and reduces temporal artifacts by leveraging warped previous-frame features, highlighting its importance for coherent video super-resolution.

Figure 9: **Ablation study on training strategy.**

Stage combination	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	MUSIQ↑	tLP↓	tOF↓	WarpErr↓
stage 1 and 2	25.442	0.702	0.156	0.100	67.528	21.781	6.37	27.307
stage 1 and 3	26.307	0.753	0.121	0.077	64.902	13.094	4.09	21.689
stage 2 and 3	26.906	0.758	0.132	0.077	64.751	10.510	4.225	15.726
All stage jointly	26.135	0.736	0.124	0.073	67.35	17.816	4.596	24.298
Sperate (Ours)	27.256	0.766	0.099	0.062	65.586	4.265	3.620	14.909

Figure 11: **Ablation study on inference steps.** The 4-step model achieves an optimal quality-efficiency trade-off compared to 1-, 10-, and 50-step variants, validating our distillation strategy.

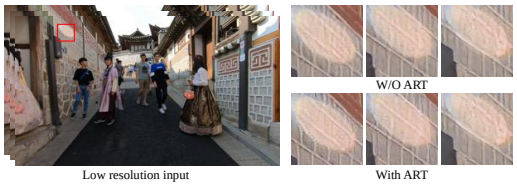

Figure 12: **Ablation study on Auto-regressive Temporal Guidance (ARTG).** ARTG enhances temporal consistency and perceptual quality by leveraging warped previous frames, reducing flickering, and improving structural coherence.

outperforms random timestep selection in both quality and efficiency, reducing training cost from 60.5 to 21 GPU hours on 4×A6000 GPUs.

To evaluate the trade-off between runtime and reconstruction quality, we vary the number of DDIM inference steps in our full Stream-DiffVSR pipeline while keeping all model weights fixed. As shown in Fig. 6 and Fig. 11, using fewer steps (e.g., 1) significantly improves efficiency but degrades perceptual quality, while more steps (e.g., 10 or 50) enhance visual fidelity at the cost of latency. Our 4-step configuration achieves the best balance, maintaining high perceptual quality under strong efficiency constraints.

Fig. 8 and Fig. 12 demonstrate the effectiveness of ARTG and TPM. The *per-frame* baseline refers to inference using only the distilled U-Net, with both ARTG and the TPM disabled. In the ablation labels, *w/o* indicates that the corresponding module is disabled entirely. For example, *TPM (unwarp)* denotes a variant where TPM receives the previous HR frame without flow-based warping, thus removing motion alignment from its temporal input. ARTG improves perceptual quality (LPIPS from 0.117 to 0.099) and temporal consistency (tLP100 from 6.132 to 4.265). TPM further enhances temporal coherence through temporal feature warping and fusion, as reflected in additional improvements in tLP100. These results highlight the complementary roles of latent-space temporal guidance and decoder-side temporal modeling.

5 Conclusion

We propose Stream-DiffVSR, an efficient online video super-resolution framework using diffusion models. By integrating a distilled U-Net, Auto-Regressive Temporal Guidance, and Temporal-aware Decoder, Stream-DiffVSR achieves superior perceptual quality, temporal consistency, and practical inference speed for low-latency applications.

Limitations. Stream-DiffVSR remains computationally heavier than CNN or Transformer methods. Optical flow reliance may introduce artifacts in fast-motion scenes, suggesting alternative motion models. Its auto-regressive design yields lower quality initial frames, indicating a need for better initialization (visual results are provided in the appendix). Improving generalization to real-world degradations also warrants further study.

6 Ethics Statement

This work on video super-resolution raises several ethical considerations. While intended for beneficial applications like improving video accessibility and conferencing quality, we acknowledge that the technology could potentially be misused for deceptive content creation or surveillance enhancement. Our training requires significant computational resources (1.26M iterations on 4× A6000 GPUs), which has environmental implications; we will release pre-trained models to prevent redundant training. The training datasets (REDS, Vimeo-90K, YouHQ) may contain geographic and demographic biases that could affect performance across different groups. We transparently disclose our method's limitations, including first-frame quality degradation and fidelity trade-offs, to ensure informed deployment decisions. We have no conflicts of interest to declare.

7 Reproducibility Statement

We provide complete implementation details for reproducibility. Sec. 3.5 and Appendix A.3 detail all training hyperparameters: learning rate (5e-5), batch size (16), optimizer (AdamW, $\beta_1 = 0.9$, $\beta_2 = 0.999$), iterations per stage (600K/600K/60K), and loss weights. Algorithms 1 and 2 in the Appendix provide explicit pseudocode. Our architecture modifications to StableVSR and AutoEncoderTiny are described in Sec. 3.4 and the Appendix A.2. We use publicly available datasets with standard splits (Appendix A.1), RAFT for optical flow, and 512×512 training patches. Evaluation uses standard metric implementations. We commit to releasing all code, trained models, and evaluation scripts upon acceptance. Training requires 4× A6000 GPUs. Inference runs on RTX 4090 with 24GB memory for 720p video.

References

Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6228–6237, 2018.

Yanpeng Cao, Chengcheng Wang, Changjun Song, Yongming Tang, and He Li. Real-time super-resolution system of 4k-video based on deep learning. In *2021 IEEE 32nd International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, pp. 69–76. IEEE, 2021.

Kelvin CK Chan, Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Basicvsr: The search for essential components in video super-resolution and beyond. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4947–4956, 2021.

Kelvin CK Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5972–5981, 2022a.

Kelvin CK Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. Investigating tradeoffs in real-world video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5962–5971, 2022b.

Rui Chen, Yang Mu, and Yan Zhang. High-order relational generative adversarial network for video super-resolution. *Pattern Recognition*, 146:110059, 2024.

Zheng Chen, Zichen Zou, Kewei Zhang, Xiongfei Su, Xin Yuan, Yong Guo, and Yulun Zhang. Dove: Efficient one-step diffusion model for real-world video super-resolution. *arXiv preprint arXiv:2505.16239*, 2025.

Marcos V Conde, Zhijun Lei, Wen Li, Christos Bampis, Ioannis Katsavounidis, Radu Timofte, Qing Luo, Jie Song, Linyan Jiang, Haibo Lei, et al. Aim 2024 challenge on efficient video super-resolution for av1 compressed content. In *European Conference on Computer Vision*, pp. 304–325. Springer, 2024.

Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision (ICCV)*, pp. 764–773, 2017.

Keyan Ding, Kede Ma, Shiqi Wang, and Eero P Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE transactions on pattern analysis and machine intelligence*, 44(5):2567–2581, 2020.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883, 2021.

Dario Fuoli, Shuhang Gu, and Radu Timofte. Efficient video super-resolution through recurrent latent space propagation. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 3476–3485. IEEE, 2019.

Sicheng Gao, Xuhui Liu, Bohan Zeng, Sheng Xu, Yanjing Li, Xiaoyan Luo, Jianzhuang Liu, Xiantong Zhen, and Baochang Zhang. Implicit diffusion models for continuous super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10021–10030, 2023.

Zhengyang Geng, Ashwini Pokle, William Luo, Justin Lin, and J Zico Kolter. Consistency models made easy. *arXiv preprint arXiv:2406.14548*, 2024.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Zujin Guo, Wei Li, and Chen Change Loy. Generalizable implicit motion modeling for video frame interpolation. *Advances in Neural Information Processing Systems*, 37:63747–63770, 2024.

Janghyeok Han, Gyujin Sim, Geonung Kim, Hyun-Seung Lee, Kyuha Choi, Youngseok Han, and Sunghyun Cho. Dc-vsr: Spatially and temporally consistent video super-resolution with video diffusion prior. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–11, 2025.

Jingwen He, Tianfan Xue, Dongyang Liu, Xinqi Lin, Peng Gao, Dahua Lin, Yu Qiao, Wanli Ouyang, and Ziwei Liu. Venhancer: Generative space-time enhancement for video generation. *arXiv preprint arXiv:2407.07667*, 2024.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022.

Takashi Isobe, Xu Jia, Shuhang Gu, Songjiang Li, Shengjin Wang, and Qi Tian. Video super-resolution with recurrent structure-detail network. *arXiv preprint arXiv:2008.00455*, 2020.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

Mehran Jeelani, Noshaba Cheema, Klaus Illgner-Fehns, Philipp Slusallek, Sunil Jaiswal, et al. Expanding synthetic real-world degradations for blind video super resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1199–1208, 2023.

Xiaozhong Ji, Yun Cao, Ying Tai, Chengjie Wang, Jilin Li, and Feiyue Huang. Real-world super-resolution via kernel estimation and noise injection. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 466–467, 2020.

Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. *arXiv preprint arXiv:2410.05954*, 2024.

Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5148–5157, 2021.

Chenyang Lei, Yazhou Xing, and Qifeng Chen. Blind video temporal consistency via deep video prior. *Advances in Neural Information Processing Systems*, 33:1083–1093, 2020.

Haoying Li, Yifan Yang, Meng Chang, Shiqi Chen, Huajun Feng, Zhihai Xu, Qi Li, and Yueting Chen. Srdiff: Single image super-resolution with diffusion probabilistic models. *Neurocomputing*, 479:47–59, 2022.

Wenbo Li, Xin Tao, Taian Guo, Lu Qi, Jiangbo Lu, and Jiaya Jia. Mucan: Multi-correspondence aggregation network for video super-resolution. *arXiv preprint arXiv:2007.11803*, 2020.

Xiaohui Li, Yihao Liu, Shuo Cao, Ziyan Chen, Shaobin Zhuang, Xiangyu Chen, Yinan He, Yi Wang, and Yu Qiao. Diffvsr: Enhancing real-world video super-resolution with diffusion models for advanced visual quality and temporal consistency. *arXiv preprint arXiv:2501.10110*, 2025.

Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1833–1844, 2021.

Jingyun Liang, Jiezhang Cao, Yuchen Fan, Kai Zhang, Rakesh Ranjan, Yawei Li, Radu Timofte, and Luc Van Gool. Vrt: A video restoration transformer. *arXiv preprint arXiv:2201.12288*, 2022a.

Jingyun Liang, Yuchen Fan, Xiaoyu Xiang, Rakesh Ranjan, Eddy Ilg, Simon Green, Jiezhang Cao, Kai Zhang, Radu Timofte, and Luc V Gool. Recurrent video restoration transformer with guided deformable attention. *Advances in Neural Information Processing Systems*, 35:378–393, 2022b.

Ce Liu and Deqing Sun. On bayesian adaptive video super resolution. *IEEE transactions on pattern analysis and machine intelligence*, 36(2):346–360, 2013.

Haozhe Liu, Shikun Liu, Zijian Zhou, Mengmeng Xu, Yanping Xie, Xiao Han, Juan C Pérez, Ding Liu, Kumara Kahatapitiya, Menglin Jia, et al. Mardini: Masked autoregressive diffusion for video generation at scale. *arXiv preprint arXiv:2410.20280*, 2024.

Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, et al. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *The Twelfth International Conference on Learning Representations*, 2023.

Yong Liu, Jinshan Pan, Yinchuan Li, Qingji Dong, Chao Zhu, Yu Guo, and Fei Wang. Ultravsr: Achieving ultra-realistic video super-resolution with efficient one-step diffusion space. *arXiv preprint arXiv:2505.19958*, 2025.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, pp. 1–22, 2025.

Cheng Lu et al. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *NeurIPS*, 35:5775–5787, 2022.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11461–11471, 2022.

Simian Luo et al. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023.

Chao Ma, Chih-Yuan Yang, Xiaokang Yang, and Ming-Hsuan Yang. Learning a no-reference quality metric for single-image super-resolution. *Computer Vision and Image Understanding*, 158:1–16, 2017.

Chenlin Meng, Robin Rombach, Ruiqi Gao, Diederik Kingma, Stefano Ermon, Jonathan Ho, and Tim Salimans. On distillation of guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14297–14306, 2023.

Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on image processing*, 21(12):4695–4708, 2012.

Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In *CVPR Workshops*, June 2019a.

Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In *CVPRW*, 2019b.

Jinshan Pan, Haoran Bai, Jiangxin Dong, Jiawei Zhang, and Jinhui Tang. Deep blind video super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4811–4820, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

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 (CVPR)*, pp. 10684–10695, June 2022a.

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*, pp. 10684–10695, 2022b.

Claudio Rota, Marco Buzzelli, and Joost van de Weijer. Enhancing perceptual quality in video super-resolution through temporally-consistent detail synthesis using diffusion models. In *European Conference on Computer Vision*, pp. 36–53. Springer, 2024.

Michele A Saad and Alan C Bovik. Blind quality assessment of videos using a model of natural scene statistics and motion coherency. In *2012 Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, pp. 332–336. IEEE, 2012.

Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE transactions on pattern analysis and machine intelligence*, 45(4):4713–4726, 2022.

Mehdi SM Sajjadi, Raviteja Vemulapalli, and Matthew Brown. Frame-recurrent video super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6626–6634, 2018.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

Shuwei Shi, Jinjin Gu, Liangbin Xie, Xintao Wang, Yujiu Yang, and Chao Dong. Rethinking alignment in video super-resolution transformers. *arXiv preprint arXiv:2207.08494*, 2022.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Yexing Song, Meilin Wang, Zhijing Yang, Xiaoyu Xian, and Yukai Shi. Negvsr: Augmenting negatives for generalized noise modeling in real-world video super-resolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 10705–10713, 2024.

Lingchen Sun, Rongyuan Wu, Zhiyuan Ma, Shuaizheng Liu, Qiaosi Yi, and Lei Zhang. Pixel-level and semantic-level adjustable super-resolution: A dual-lora approach. 2025a.

Mingzhen Sun, Weining Wang, Gen Li, Jiawei Liu, Jiahui Sun, Wanquan Feng, Shanshan Lao, SiYu Zhou, Qian He, and Jing Liu. Ar-diffusion: Asynchronous video generation with auto-regressive diffusion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 7364–7373, 2025b.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pp. 402–419. Springer, 2020.

Yapeng Tian, Yulun Zhang, Yun Fu, and Chenliang Xu. Tdan: Temporally-deformable alignment network for video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3360–3369, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Fu-Yun Wang, Zhaoyang Huang, Alexander Bergman, Dazhong Shen, Peng Gao, Michael Lingelbach, Keqiang Sun, Weikang Bian, Guanglu Song, Yu Liu, et al. Phased consistency models. *Advances in neural information processing systems*, 37:83951–84009, 2024a.

Fu-Yun Wang, Ling Yang, Zhaoyang Huang, Mengdi Wang, and Hongsheng Li. Rectified diffusion: Straightness is not your need in rectified flow. *arXiv preprint arXiv:2410.07303*, 2024b.

Hengkang Wang, Yang Liu, Huidong Liu, Chien-Chih Wang, Yanhui Guo, Hongdong Li, Bryan Wang, and Ju Sun. Temporal-consistent video restoration with pre-trained diffusion models. *arXiv preprint arXiv:2503.14863*, 2025.

Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 2555–2563, 2023.

Jianyi Wang, Zongsheng Yue, Shangchen Zhou, Kelvin CK Chan, and Chen Change Loy. Exploiting diffusion prior for real-world image super-resolution. *International Journal of Computer Vision*, 132(12):5929–5949, 2024c.

Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 0–0, 2019.

Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *International Conference on Computer Vision Workshops (ICCVW)*, 2021.

Shuchen Weng, Haojie Zheng, Peixuan Zhan, Yuchen Hong, Han Jiang, Si Li, and Boxin Shi. Vires: Video instance repainting with sketch and text guidance. *arXiv preprint arXiv:2411.16199*, 2024.

Rongyuan Wu, Lingchen Sun, Zhiyuan Ma, and Lei Zhang. One-step effective diffusion network for real-world image super-resolution. *arXiv preprint arXiv:2406.08177*, 2024a.

Rongyuan Wu, Tao Yang, Lingchen Sun, Zhengqiang Zhang, Shuai Li, and Lei Zhang. Seesr: Towards semantics-aware real-world image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 25456–25467, 2024b.

Desai Xie, Zhan Xu, Yicong Hong, Hao Tan, Difan Liu, Feng Liu, Arie Kaufman, and Yang Zhou. Progressive autoregressive video diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 6322–6332, 2025.

Sirui Xie, Zhisheng Xiao, Diederik Kingma, Tingbo Hou, Ying Nian Wu, Kevin P Murphy, Tim Salimans, Ben Poole, and Ruiqi Gao. Em distillation for one-step diffusion models. *Advances in Neural Information Processing Systems*, 37:45073–45104, 2024.

Yiran Xu, Taesung Park, Richard Zhang, Yang Zhou, Eli Shechtman, Feng Liu, Jia-Bin Huang, and Difan Liu. Videogigagan: Towards detail-rich video super-resolution. 2024.

Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8):1106–1125, 2019.

Xi Yang, Wangmeng Xiang, Hui Zeng, and Lei Zhang. Real-world video super-resolution: A benchmark dataset and a decomposition based learning scheme. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4781–4790, 2021.

Xi Yang, Chenhang He, Jianqi Ma, and Lei Zhang. Motion-guided latent diffusion for temporally consistent real-world video super-resolution. In *European Conference on Computer Vision*, pp. 224–242. Springer, 2024.

Peng Yi, Zhongyuan Wang, Kui Jiang, Junjun Jiang, and Jiayi Ma. Progressive fusion video super-resolution network via exploiting non-local spatio-temporal correlations. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3106–3115, 2019.

Geunhyuk Youk, Jihyong Oh, and Munchurl Kim. Fma-net: Flow-guided dynamic filtering and iterative feature refinement with multi-attention for joint video super-resolution and deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 44–55, 2024.

Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5728–5739, 2022.

Aiping Zhang, Zongsheng Yue, Renjing Pei, Wenqi Ren, and Xiaochun Cao. Degradation-guided one-step image super-resolution with diffusion priors. *arxiv*, 2024a.

Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. Designing a practical degradation model for deep blind image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4791–4800, 2021.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018.

Ruofan Zhang, Jinjin Gu, Haoyu Chen, Chao Dong, Yulun Zhang, and Wenming Yang. Crafting training degradation distribution for the accuracy-generalization trade-off in real-world super-resolution. 2023.

Yuehan Zhang and Angela Yao. Realviformer: Investigating attention for real-world video super-resolution. In *European Conference on Computer Vision*, pp. 412–428. Springer, 2024.

Zhengqiang Zhang, Ruihuang Li, Shi Guo, Yang Cao, and Lei Zhang. Tmp: Temporal motion propagation for online video super-resolution. *IEEE Transactions on Image Processing*, 2024b.

Zhixing Zhang, Bichen Wu, Xiaoyan Wang, Yaqiao Luo, Luxin Zhang, Yinan Zhao, Peter Vajda, Dimitris Metaxas, and Licheng Yu. Avid: Any-length video inpainting with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7162–7172, 2024c.

Weisong Zhao, Jingkai Zhou, Xiangyu Zhu, Weihua Chen, Xiao-Yu Zhang, Zhen Lei, and Fan Wang. Realisvsr: Detail-enhanced diffusion for real-world 4k video super-resolution. *arXiv preprint arXiv:2507.19138*, 2025.

Kaiwen Zheng, Cheng Lu, Jianfei Chen, and Jun Zhu. Dpm-solver-v3: Improved diffusion ode solver with empirical model statistics. *Advances in Neural Information Processing Systems*, 36: 55502–55542, 2023.

Shangchen Zhou, Peiqing Yang, Jianyi Wang, Yihang Luo, and Chen Change Loy. Upscale-a-video: Temporal-consistent diffusion model for real-world video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2535–2545, 2024a.

Xingyu Zhou, Leheng Zhang, Xiaorui Zhao, Keze Wang, Leida Li, and Shuhang Gu. Video super-resolution transformer with masked inter&intra-frame attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25399–25408, 2024b.

Zhenyu Zhou, Defang Chen, Can Wang, Chun Chen, and Siwei Lyu. Simple and fast distillation of diffusion models. *Advances in Neural Information Processing Systems*, 37:40831–40860, 2024c.

Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9308–9316, 2019.

A APPENDIX

A.1 EXPERIMENTAL SETUP

Training and Evaluation Setup. Stream-DiffVSR is trained in three sequential stages to ensure stable optimization and modular control over temporal components. All evaluation experiments are conducted on an NVIDIA RTX 4090 GPU with TensorRT acceleration. Details of the stage-wise training procedure and configurations are provided in the appendix.

Datasets. We evaluate our method using widely-recognized benchmarks: REDS (Nah et al., 2019b), Vimeo-90K (Xue et al., 2019), and YouHQ (Zhou et al., 2024a). REDS consists of 300 video sequences (1280×720 resolution, 100 frames each); sequences 000, 011, 015, and 020 (REDS4) are used for testing. Vimeo-90K-T contains 91,701 clips (448×256 resolution), with 64,612 for training and 7,824 (Vimeo-90K) for evaluation. YouHQ provides 38,576 videos high-resolution YouTube clips (up to 1080p) for training, offering diverse real-world content for training.

For testing under real-world degradation, we also evaluate on two additional benchmarks: VideoLQ (Zhang et al., 2021), a no-reference video quality dataset curated from real Internet content, and Vid4 (Liu & Sun, 2013), a classical benchmark with 4 videos commonly used for VSR evaluation. The evaluation results are provided in appendix.

Evaluation metrics. We assess the effectiveness of our approach using a comprehensive set of perceptual and temporal metrics across multiple aspects. **Reference-based Perceptual Quality**: LPIPS (Zhang et al., 2018) and DISTS (Ding et al., 2020). **No-reference Perceptual Quality**: MUSIQ (Ke et al., 2021), NIQE (Saad & Bovik, 2012), CLIP-IQA (Wang et al., 2023), NRQM (Ma et al., 2017), BRISQUE (Mittal et al., 2012). **Temporal Consistency**: Temporal Learned Perceptual Similarity (tLP), and Temporal Optical Flow difference (tOF). **Inference Speed**: Per-frame runtime measured on an NVIDIA RTX 4090 GPU to evaluate low-latency applicability. Note that while we report PSNR and SSIM results (REDS4: 27.256 / 0.768) for completeness, we do not rely on these distortion-based metrics in our main analysis, as they often fail to reflect perceptual quality and temporal coherence, especially in generative VSR settings. This has also been observed in prior work (Zhang et al., 2018). Our qualitative results demonstrate superior perceptual and temporal quality, as we prioritize low-latency stability and consistency over overfitting to any single metric.

Baseline methods. We evaluate our method against leading CNN-based, Transformer-based, and Diffusion-based models. Specifically, we include bidirectional (offline) methods such as BasicVSR++(Chan et al., 2022a), RealBasicVSR(Chan et al., 2022b), RVRT (Liang et al., 2022b), StableVSR (Rota et al., 2024), MGLD-VSR (Yang et al., 2024), and unidirectional (online) methods including MIA-VSR (Zhou et al., 2024b), TMP (Zhang et al., 2024b), RealViformer (Zhang & Yao, 2024), and StableVSR* (Rota et al., 2024), comprehensively comparing runtime, perceptual quality, and temporal consistency.

A.2 ADDITIONAL IMPLEMENTATION DETAILS

Implementation Details. Our UNet backbone is initialized from the StableVSR (Rota et al., 2024) released UNet checkpoint, which is trained for image-based super-resolution from Stable Diffusion (SD) x4 Upscaler (Rombach et al., 2022b;a). We then perform 4-step distillation to adapt this UNet for efficient video SR. ARTG, in contrast, is built upon our distilled UNet encoder and computes temporal residuals from previous high-resolution outputs using convolutional and transformer blocks. These residuals are injected into the decoder during upsampling, enhancing temporal consistency without modifying the encoder or increasing diffusion steps. Our decoder is initialized from AutoEncoderTiny and extended with a Temporal Processor Module (TPM) to incorporate multi-scale temporal fusion during final reconstruction.

A.3 ADDITIONAL TRAINING DETIALS

Stage 1: U-Net Distillation. We initialize the denoising U-Net from the 50-step diffusion model released by StableVSR (Rota et al., 2024), which was trained on REDS (Nah et al., 2019a) dataset. To ac-

[0]*StableVSR (Rota et al., 2024) is originally a bidirectional model. We implement a unidirectional variant (StableVSR*) that only uses forward optical flow for fair comparison under the online setting.

celerate inference, we distill the 50-step U-Net into a 4-step variant using a deterministic DDIM (Song et al., 2020) scheduler. During training, our rollout distillation always starts from the noisiest latent at timestep 999 and executes the full sequence of four denoising steps $\{999, 749, 499, 249\}$. Supervision is applied only to the final denoised latent at $t = 0$, ensuring that training strictly mirrors the inference trajectory and reducing the gap between training and inference. We use a batch size of 16, learning rate of 5e-5 with constant, and AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 0.01). Training is conducted for 600K iterations with a patch size of 512×512.The distillation loss consists of MSE loss in latent space, LPIPS (Zhang et al., 2018) loss, and adversarial loss using a PatchGAN discriminator (Isola et al., 2017) in pixel level, with weights of 1.0, 0.5, and 0.025 respectively. Adversarial loss are envolved after 20k iteration for training stabilization.

Stage 2: Temporal-aware Decoder Training. The decoder receives both the encoded ground truth latent features and temporally aligned context features (via flow-warped previous frames). The encoder used to extract temporal features is frozen.We use a batch size of 16, learning rate of 5e-5 with constant, and AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 0.01). Training is conducted for 600K iterations with a patch size of 512×512. Loss consists of smooth L1 reconstruction loss, LPIPS (Zhang et al., 2018) loss, flow loss using RAFT (Teed & Deng, 2020) and adversarial loss using a PatchGAN discriminator (Isola et al., 2017) in pixel level for training, with weights of 1.0, 0.3, 0.1 and 0.025 respectively. Flow loss and adversarial loss are envolved after 20k iteration for training stabilization.

Stage 3: Auto-regressive Temporal Guidance. We train the ARTG module while freezing both the U-Net and decoder. Optical flow is computed between adjacent frames using RAFT (Teed & Deng, 2020), and the warped previous super-resolved frame is injected into the denoising U-Net and decoder. The loss formulation is identical to Stage 1, conducted with 60K iterations. This guides ARTG to enhance temporal coherence while maintaining alignment with the original perceptual objectives.

Algorithm 1: Training procedure for U-Net rollout distillation.

Input: Dataset $\mathcal{D} = \{(\tilde{I}, I)\}$; pre-trained VAE; 4-step noise scheduler; student U-Net with parameters θ; discriminator $D(\cdot)$.

for *epoch* $= 1$ *to* N **do**
  for *each batch* $(\tilde{I}, I) \in \mathcal{D}$ **do**
    $\mathbf{z}_0 \leftarrow \mathrm{VAE.encode}(I)$;
    Sample $\epsilon \sim \mathcal{N}(0, I)$;
    $\mathbf{z}_T \leftarrow \alpha_T \mathbf{z}_0 + \sqrt{1 - \alpha_T}\,\epsilon$; `// Add noise at maximum timestep T`
    `// -- Rollout 4-step denoising --`
    $\hat{\mathbf{z}}_T \leftarrow [\mathbf{z}_T, \tilde{I}]$;
    for *step* $s = T, \ldots, 1$ **do**
      $\hat{\epsilon} \leftarrow \mathrm{U{-}Net}(\hat{\mathbf{z}}_s, s)$;
      $\hat{\mathbf{z}}_{s-1} \leftarrow \mathrm{Scheduler.step}(\hat{\epsilon}, s, \hat{\mathbf{z}}_s)$;
    $\hat{I} \leftarrow \mathrm{VAE.decode}(\hat{\mathbf{z}}_0)$;
    $\mathcal{L}_{\mathrm{L2}} \leftarrow \|\hat{I} - I\|_2^2$;
    $\mathcal{L}_{\mathrm{LPIPS}} \leftarrow \mathrm{LPIPS}(\hat{I}, I)$;
    $\mathcal{L}_{\mathrm{GAN}} \leftarrow \mathrm{softplus}\big(-D(\hat{I})\big)$;
    $\mathcal{L} \leftarrow \lambda_{L2}\,\mathcal{L}_{\mathrm{L2}} + \lambda_{\mathrm{LPIPS}}\,\mathcal{L}_{\mathrm{LPIPS}} + \lambda_{\mathrm{GAN}}\,\mathcal{L}_{\mathrm{GAN}}$;
    Update parameters: $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$;

A.4 Additional Quantitative comparison.

We provide extended quantitative results across multiple datasets and settings. Specifically, we report both bidirectional and unidirectional performance with mean and standard deviation on REDS4 (Tables 6 and 7) and Vimeo-90K (Tables 8 and 9), while additional bidirectional results are provided on VideoLQ (Tables 10 and 11) and Vid4 (Tables 12 and 13). These supplementary results further validate the robustness of our approach under diverse benchmarks and temporal settings.

Algorithm 2: Auto-Regressive Diffusion VSR.

Notation: $\{\tilde{I}_i\}$: Input LR frames, $\{\hat{I}_i\}$: Enhanced frames, FlowWarp: Warping w.r.t. flow, VAE: Auto-regressive VAE, UNet: Distilled diffusion U-Net, ARTG: Auto-Regressive Temporal guidance, PrepareLatents: Create latent input, timesteps: $\{t_1, \ldots, t_4\}$
Input: $\{\tilde{I}_i\}_{i=1}^{N}$, flows $\{\mathbf{f}_{i-1}\}_{i=2}^{N}$, VAE, UNet, ARTG.
Output: $\{\hat{I}_i\}_{i=1}^{N}$.

for $i = 1$ **to** N **do**
  $\mathbf{LQ}_i \leftarrow \tilde{I}_i$
  $\mathbf{z}_i \leftarrow \text{PrepareLatents}(\mathbf{LQ}_i,\ t)$
  if $i > 1$ **then**
    $\hat{I}^w_{i-1} \leftarrow \text{FlowWarp}(\hat{I}_{i-1},\ \mathbf{f}_{i-1})$
    $\mathbf{E}_{i-1} \leftarrow \text{VAE.encode}(\hat{I}^w_{i-1})$
  for $t \in$ timesteps **do**
    if $i > 1$ **then**
      $\mathbf{z}_i \leftarrow \text{ARTG}(\mathbf{z}_i,\ \hat{I}^w_{i-1})$
    $\hat{\epsilon} \leftarrow \text{UNet}(\mathbf{z}_i,\ t)$
    $\mathbf{z}_i \leftarrow \text{DiffusionUpdate}(\hat{\epsilon},\ t,\ \mathbf{z}_i)$
  if $i > 1$ **then**
    $\hat{I}_i \leftarrow \text{VAE.Decode}(\mathbf{z},\ \mathbf{E}_{i-1})$
  else
    $\hat{I}_i \leftarrow \text{VAE.Decode}(\mathbf{z})$
return $\{\hat{I}_i\}$

A.5 Additional Visual Result

Figure Figs. 13 to 15 presents qualitative results on challenging real-world sequences. Compared with CNN-based (TMP, BasicVSR++) and Transformer-based (RealViFormer) approaches, as well as diffusion-based MGLD-VSR, our method produces sharper structures and more faithful textures while reducing temporal flickering. These visual comparisons further demonstrate the effectiveness of our design in maintaining perceptual quality and temporal consistency across diverse scenes.

A.6 Failure Cases

Figure Fig. 19 illustrates a limitation of our approach on the first frame of a video sequence. Since no past frames are available for temporal guidance, the model may produce blurrier details or less stable structures compared to subsequent frames. This issue is inherent to all online VSR settings, where temporal information cannot be exploited at the sequence start. As shown in later frames, once temporal context becomes available, our method quickly stabilizes and reconstructs high-fidelity details.

Table 6: **Quantitative comparison against bidirectional/offline methods on the REDS4 dataset.** We compare CNN-, Transformer-, and diffusion-based methods on REDS4. Stream-DiffVSR achieves superior perceptual and temporal quality with high stability across sequences. All values are *mean* ± *std* over 4 videos. ↑ indicates higher is better; ↓ indicates lower is better. **Dir.** denotes temporal direction: **B** for bidirectional/offline, **U** for unidirectional/online. Runtime is measured per 720p frame on an RTX 4090.**Latency-first** measures first frame latency); **Latency-avg** is the average per-frame latency across the entire sequence. **tLP** and **tOF** are scaled by 100× and 10×. Best and second-best results are marked in red and blue.

Dir.	Method	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	MUSIQ↑	NIQE↓	NRQM↑	BRISQUE↓	tLP↓	tOF↓	Runtime (s)↓	latency-first (s)↓	latency-avg (s)↓
CNN-based Methods														
-	Bicubic	25.501 ± 1.516	0.712 ± 0.062	0.460 ± 0.042	0.187 ± 0.013	27.362 ± 2.239	7.360 ± 0.120	3.459 ± 0.177	60.256 ± 1.828	21.603 ± 5.817	4.241 ± 5.765	-	-	-
B	BasicVSR++	32.386 ± 2.415	0.907 ± 0.029	0.132 ± 0.023	0.069 ± 0.012	67.002 ± 4.291	3.850 ± 0.439	6.363 ± 0.330	38.641 ± 5.224	9.017 ± 4.384	2.490 ± 4.440	0.098	9.8	4.9
B	RealBasicVSR	27.042 ± 1.865	0.778 ± 0.059	0.134 ± 0.016	0.060 ± 0.006	67.033 ± 4.283	2.530 ± 0.452	6.769 ± 0.242	18.046 ± 4.185	6.422 ± 4.726	4.759 ± 7.722	0.064	6.4	3.2
Transformer-based Methods														
B	RVRT	32.701 ± 2.487	0.911 ± 0.027	0.130 ± 0.022	0.067 ± 0.011	67.251 ± 4.372	3.793 ± 0.463	6.366 ± 0.339	38.038 ± 5.779	9.133 ± 4.408	2.421 ± 4.316	0.498	49.8	24.9
B	MIA-VSR	32.790 ± 2.535	0.912 ± 0.028	0.123 ± 0.022	0.064 ± 0.011	68.140 ± 3.964	3.742 ± 0.472	6.451 ± 0.304	37.099 ± 5.668	8.870 ± 4.606	2.354 ± 4.026	0.768	0.768	0.768
Diffusion-based Methods														
B	StableVSR	27.928 ± 2.411	0.793 ± 0.063	0.102 ± 0.015	0.047 ± 0.006	67.058 ± 3.797	2.713 ± 0.456	6.960 ± 0.211	16.249 ± 4.133	5.755 ± 4.618	2.742 ± 4.741	46.2	4620	2310
B	MGLD-VSR	26.53 ± 1.939	0.749 ± 0.062	0.151 ± 0.019	0.065 ± 0.006	66.081 ± 4.027	2.972 ± 0.386	6.701 ± 0.202	15.291 ± 4.463	18.139 ± 8.772	5.910 ± 6.888	43.6	218	109
U	Ours	27.256 ± 2.134	0.766 ± 0.062	0.099 ± 0.013	0.062 ± 0.007	65.595 ± 3.982	3.114 ± 0.186	7.055 ± 0.257	17.117 ± 1.836	4.198 ± 3.795	3.638 ± 4.855	0.328	0.328	0.328

Table 7: **Quantitative comparison against unidirectional/online methods on the REDS4 dataset.**

Dir.	Method	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	MUSIQ↑	NIQE↓	NRQM↑	BRISQUE↓	tLP↓	tOF↓	Runtime (s)↓	latency-first (s)↓	latency-avg (s)↓
CNN-based Methods														
-	Bicubic	25.501 ± 1.516	0.712 ± 0.062	0.460 ± 0.042	0.187 ± 0.013	27.362 ± 2.239	7.360 ± 0.120	3.459 ± 0.177	60.256 ± 1.828	21.603 ± 5.817	4.241 ± 5.765	-	-	-
U	TMP	30.672 ± 2.317	0.871 ± 0.039	0.194 ± 0.039	0.090 ± 0.010	63.818 ± 4.129	4.378 ± 0.333	5.796 ± 0.312	43.394 ± 4.442	10.424 ± 5.654	2.480 ± 3.852	0.041	0.041	0.041
Transformer-based Methods														
U	RealViformer	26.763 ± 1.898	0.761 ± 0.062	0.129 ± 0.062	0.065 ± 0.004	64.585 ± 5.117	2.731 ± 0.454	6.356 ± 0.079	17.272 ± 4.546	11.261 ± 5.613	11.782 ± 3.762	0.099	9.9	4.95
Diffusion-based Methods														
U	StableVSR*	27.174 ± 2.449	0.763 ± 0.069	0.111 ± 0.017	0.051 ± 0.006	66.428 ± 4.040	2.572 ± 0.356	6.944 ± 0.211	15.805 ± 4.626	11.107 ± 8.293	3.925 ± 4.561	46.2	4620	2310
U	Ours	27.256 ± 2.134	0.766 ± 0.062	0.099 ± 0.013	0.062 ± 0.007	65.595 ± 3.982	3.114 ± 0.186	7.055 ± 0.257	17.117 ± 1.836	4.198 ± 3.795	3.638 ± 4.855	0.328	0.328	0.328

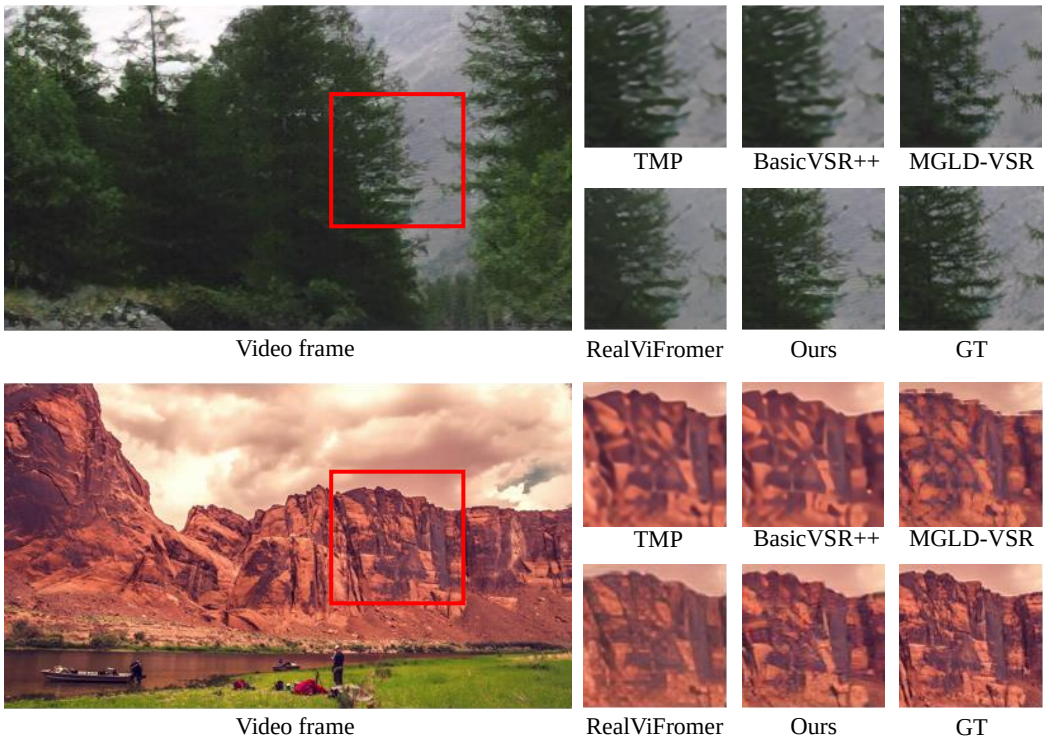

Figure 13: **Additional visual results.**

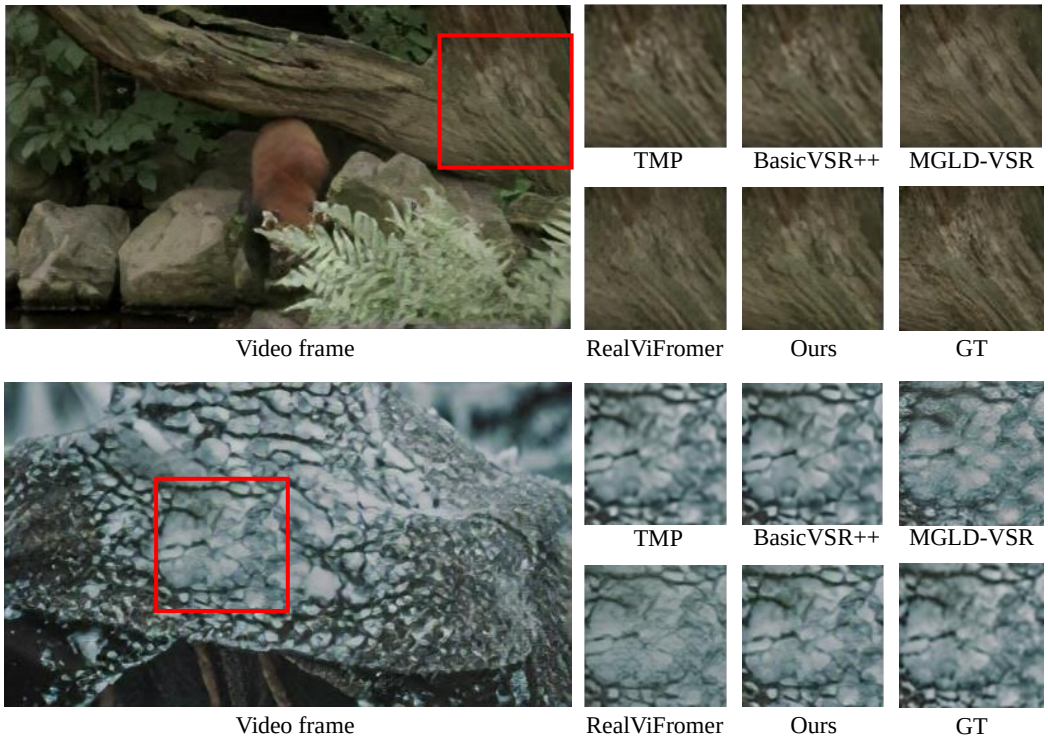

Figure 14: **Additional visual results.**

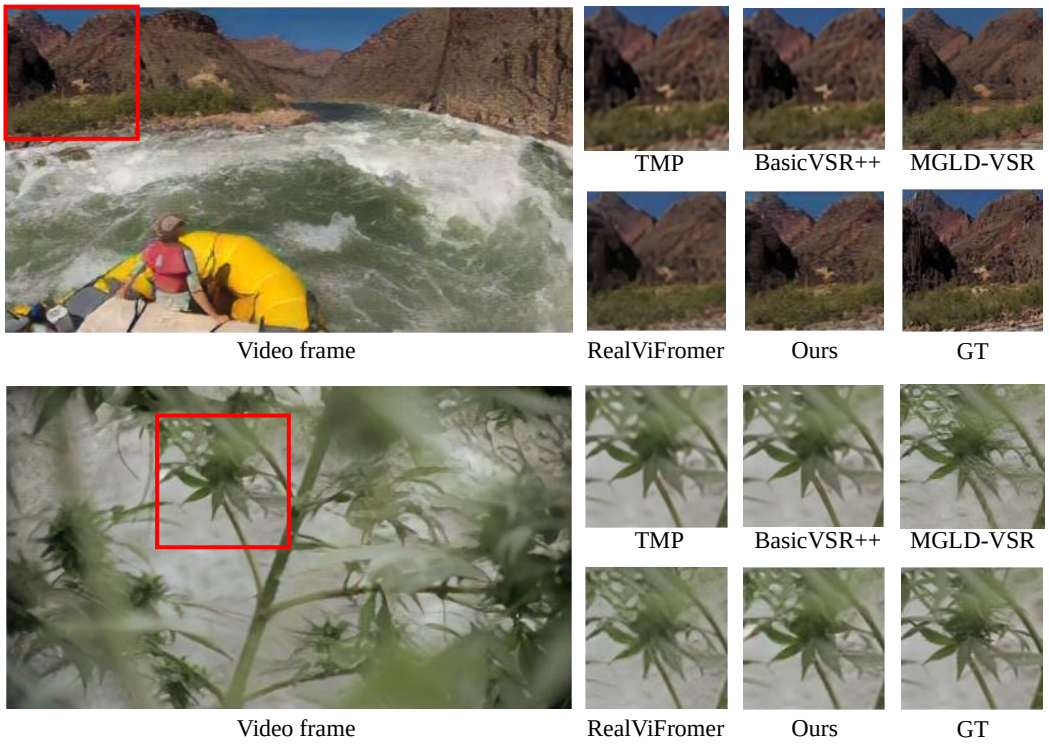

Figure 15: **Additional visual results.**

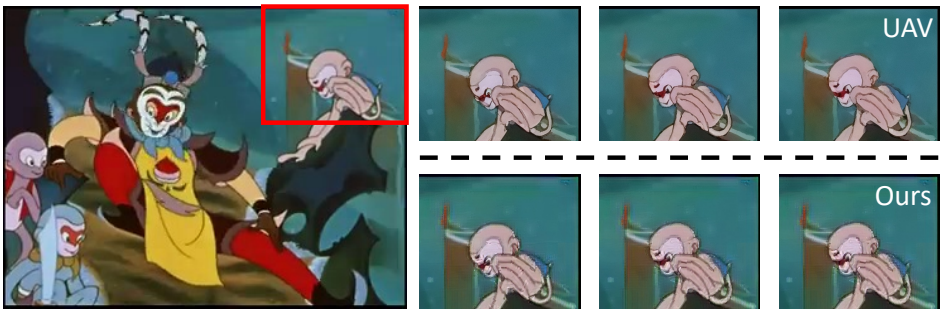

Figure 16: **Qualitative comparison with Upscale-A-Video (UAV) Zhou et al. (2024a).** Due to GPU memory limitations (OOM on an RTX 4090), we use UAV results extracted from its official project video for qualitative comparison. Despite this constraint, our Stream-DiffVSR exhibits superior visual fidelity and temporal consistency across frames.

Figure 17: **Temporal consistency comparison.** Qualitative comparison of temporal consistency across consecutive frames. Our proposed Stream-DiffVSR effectively mitigates flickering artifacts and maintains stable texture reconstruction, demonstrating superior temporal coherence compared to existing VSR methods.

A.7 Use of Large Language Models.

Large language models (LLMs) were used solely as writing assistants, for example, to improve grammar, clarity of exposition, and formatting. No part of the research idea, experimental design, implementation, or analysis was generated by LLMs. The authors take full responsibility for all scientific content of this paper.

Table 8: **Quantitative comparison on the Vimeo-90K-T dataset(bidirectional/offline).** Our Stream-DiffVSR consistently outperforms other unidirectional methods in perceptual quality and temporal consistency, while also demonstrating substantially lower runtime. All results are reported as *mean* ± *standard deviation* across the Vimeo-90K-T dataset. Runtime denotes the average per-frame inference time (in seconds) on 448×256 resolution videos using a single NVIDIA RTX 4090 GPU. Results are measured using the official implementations where available. Best and second-best results are highlighted in red and blue, respectively.

Dir.	Method	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	MUSIQ↑	NIQE↓	NRQM↑	BRISQUE↓	tLP↓	tOF↓	Runtime (s)↓	latency-first (s)↓	latency-avg (s)↓
CNN-based Methods														
-	Bicubic	29.282 ± 3.647	0.864 ± 0.061	0.297 ± 0.105	0.209 ± 0.044	23.433 ± 5.633	8.735 ± 0.397	3.588 ± 0.43	61.714 ± 4.599	11.606 ± 7.674	2.49 ± 1.645	-	-	-
B	BasicVSR++	37.479 ± 4.724	0.956 ± 0.033	0.098 ± 0.04	0.117 ± 0.024	51.940 ± 6.169	7.077 ± 1.111	5.509 ± 3.514	47.792 ± 12.514	4.691 ± 5.013	1.57 ± 0.974	0.012	0.084	0.042
B	RealBasicVSR	29.388 ± 2.692	0.857 ± 0.059	0.156 ± 0.113	0.149 ± 0.06	56.986 ± 4.418	5.069 ± 0.464	7.413 ± 0.66	23.822 ± 10.19	10.947 ± 14.292	3.46 ± 2.446	0.008	0.056	0.028
Transformer-based Methods														
B	RVRT	37.815 ± 5.049	0.955 ± 0.033	0.093 ± 0.05	0.105 ± 0.023	49.937 ± 6.509	7.205 ± 1.005	5.393 ± 0.992	48.352 ± 12.147	4.873 ± 6.486	1.429 ± 1.079	0.061	0.427	0.213
B	MIA-VSR	37.598 ± 4.724	0.957 ± 0.032	0.086 ± 0.039	0.101 ± 0.025	51.402 ± 6.522	7.116 ± 1.158	5.569 ± 1.249	47.865 ± 13.17	4.696 ± 5.874	1.419 ± 0.997	0.096	0.096	0.096
Diffusion-based Methods														
B	StableVSR	31.823 ± 3.686	0.878 ± 0.058	0.095 ± 0.044	0.111 ± 0.025	54.582 ± 6.111	4.745 ± 0.857	7.265 ± 1.427	20.039 ± 6.398	26.224 ± 9.042	3.108 ± 2.794	5.749	40.243	20.121
B	MGLD-VSR	29.651 ± 2.354	0.865 ± 0.057	0.151 ± 0.076	0.137 ± 0.032	57.788 ± 3.876	5.340 ± 0.798	7.217 ± 0.814	20.761 ± 8.394	12.550 ± 10.504	4.661 ± 3.449	5.426	27.130	13.560
U	Ours	32.593 ± 3.82	0.900 ± 0.060	0.056 ± 0.035	0.105 ± 0.017	52.755 ± 6.017	4.403 ± 1.02	7.672 ± 1.476	29.297 ± 10.007	4.307 ± 4.359	2.689 ± 1.619	0.041	0.041	0.041

Figure 18: **Optical flow visualization comparison.** Visualization of optical flow consistency across different VSR methods. Our proposed Stream-DiffVSR produces smoother and more temporally coherent flow fields, indicating improved motion consistency and reduced temporal artifacts compared to competing approaches.

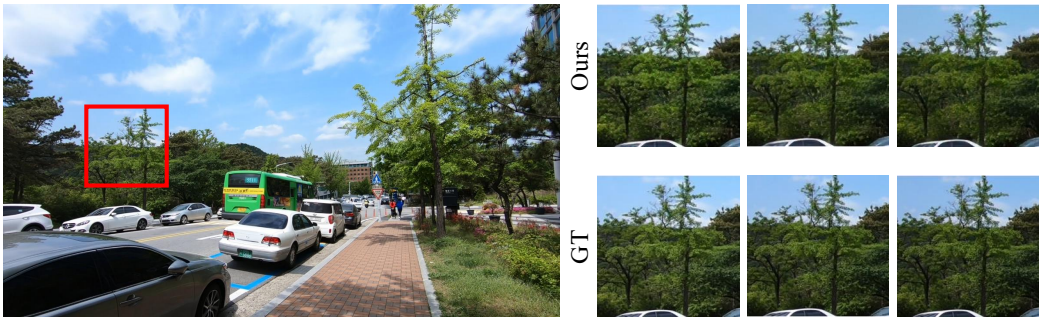

Figure 19: **Limitation on the first frame without temporal context.** Our method may underperform on the first frame of a video sequence due to the absence of prior temporal information. This limitation is inherent to online VSR settings, where no past frames are available for guidance.

Table 9: **Quantitative comparison on the Vimeo-90K-T dataset(unidirectional/online).** Our Stream-DiffVSR consistently outperforms other unidirectional methods in perceptual quality and temporal consistency, while also demonstrating substantially lower runtime. All results are reported as *mean* ± *standard deviation* across the Vimeo-90K-T dataset. Runtime denotes the average per-frame inference time (in seconds) on 448×256 resolution videos using a single NVIDIA RTX 4090 GPU. Results are measured using the official implementations where available. Best and second-best results are highlighted in red and blue, respectively.

Dir.	Method	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	MUSIQ↑	NIQE↓	NRQM↑	BRISQUE↓	tLP↓	tOF↓	Runtime (s)↓	latency-first (s)↓	latency-avg (s)↓
CNN-based Methods														
-	Bicubic	29.282 ± 3.647	0.864 ± 0.061	0.297 ± 0.105	0.209 ± 0.044	23.433 ± 5.633	8.735 ± 0.397	3.588 ± 0.43	61.714 ± 4.599	11.606 ± 7.674	2.49 ± 1.645	-	-	-
U	TMP	36.482 ± 4.672	0.946 ± 0.039	0.109 ± 0.057	0.118 ± 0.027	48.374 ± 6.31	7.368 ± 0.909	5.096 ± 0.891	49.192 ± 11.55	4.870 ± 5.177	1.603 ± 1.011	0.006	0.006	0.006
Transformer-based Methods														
U	RealViformer	30.291 ± 2.518	0.877 ± 0.055	0.130 ± 0.061	0.140 ± 0.03	53.107 ± 3.65	5.515 ± 0.486	6.711 ± 0.889	24.628 ± 7.933	8.232 ± 6.864	2.769 ± 1.909	0.013	0.091	0.045
Diffusion-based Methods														
U	StableVSR*	31.729 ± 3.698	0.875 ± 0.061	0.098 ± 0.049	0.113 ± 0.026	54.447 ± 6.008	4.698 ± 0.853	7.280 ± 1.444	19.836 ± 6.131	30.858 ± 13.166	3.144 ± 2.845	5.749	40.243	20.121
U	Ours	32.593 ± 3.82	0.900 ± 0.060	0.056 ± 0.035	0.105 ± 0.017	52.755 ± 6.017	4.403 ± 1.02	7.672 ± 1.476	29.297 ± 10.007	4.307 ± 4.359	2.689 ± 1.619	0.041	0.041	0.041

Table 10: **Quantitative comparison against bidirectional/offline methods on the VideoLQ dataset.**

Dir.	Method	NIQE↓	NRQM↑	BRISQUE↓
CNN-based Methods				
-	Bicubic	7.945	3.151	57.944
B	BasicVSR++	5.909	3.745	56.800
B	RealBasicVSR	3.973	6.095	30.158
Transformer-based Methods				
B	RVRT	6.939	3.493	60.557
B	MIA-VSR	5.860	3.810	58.513
Diffusion-based Methods				
B	StableVSR	3.973	6.154	22.973
B	MGLD-VSR	4.163	5.761	29.497
U	Ours	3.929	6.140	23.176

Table 11: **Quantitative comparison against unidirectional/online methods on the VideoLQ dataset.**

Dir.	Method	NIQE↓	NRQM↑	BRISQUE↓
CNN-based Methods				
-	Bicubic	7.945	3.151	57.944
U	TMP	6.751	3.511	59.841
Transformer-based Methods				
U	RealViformer	4.070	6.066	28.266
Diffusion-based Methods				
U	StableVSR*	3.982	6.122	23.814
U	Ours	3.929	6.140	23.176

Table 12: **Quantitative comparison against bidirectional/offline methods on the Vid4 dataset.**

Dir.	Method	PSNR↑	SSIM↑	LPIPS↓	NRQM↑	BRISQUE↓	tLP↓	tOF↓	latency-max (s)↓
CNN-based Methods									
-	Bicubic	21.719	0.582	0.512	3.429	58.680	27.819	1.145	-
B	BasicVSR++	26.230	0.828	0.193	6.481	38.409	15.029	0.507	6.86
B	RealBasicVSR	21.963	0.597	0.210	7.122	21.804	6.630	0.9	4.48
Transformer-based Methods									
B	RVRT	26.377	0.826	0.229	6.006	44.667	17.146	0.507	1.743
B	MIA-VSR	26.175	0.826	0.174	6.619	38.509	14.297	0.505	53.76
Diffusion-based Methods									
B	StableVSR	22.541	0.644	0.194	7.224	13.254	48.585	0.957	3234
B	MGLD-VSR	21.983	0.605	0.243	7.129	16.525	31.744	3.152	152.6
U	Ours	22.725	0.652	0.191	7.346	15.260	8.985	0.962	0.229

Table 13: **Quantitative comparison against unidirectional/online methods on the Vid4 dataset.**

Dir.	Method	PSNR↑	SSIM↑	LPIPS↓	NRQM↑	BRISQUE↓	tLP↓	tOF↓	latency-max (s)↓
CNN-based Methods									
-	Bicubic	21.719	0.582	0.512	3.429	58.680	27.819	1.145	-
U	TMP	25.579	0.797	0.256	5.698	46.257	14.199	0.566	0.029
Transformer-based Methods									
U	RealViformer	21.963	0.597	0.257	7.604	21.804	11.633	1.107	6.93
Diffusion-based Methods									
U	StableVSR*	22.213	0.623	0.203	7.233	11.966	59.594	1.036	3234
U	Ours	22.725	0.652	0.191	7.346	15.260	8.985	0.962	0.229