Wavelet-Space Representations for Neural Super-Resolution in Rendering Pipelines⋆

Prateek Poudel[a,*,1], Prashant Aryal[a,1], Kirtan Kunwar[a,1], Navin Nepal[a,1] and Dinesh Baniya Kshatri[a]

[a]Department of Electronics and Computer Engineering, Thapathali Campus, Tribhuwan University, , Kathmandu, 46000, Nepal

ARTICLE INFO

Keywords:
G-buffers
Stationary Wavelet transform
Super-Resolution
Neural networks
Rendering pipelines

ABSTRACT

We investigate the use of wavelet-space feature decomposition in neural super-resolution for rendering pipelines. Building on recent neural upscaling frameworks, we introduce a formulation that predicts stationary wavelet coefficients rather than directly regressing RGB values. This frequency-aware decomposition separates low- and high-frequency components, enabling sharper texture recovery and reducing blur in challenging regions. Unlike conventional wavelet transforms, our use of the stationary wavelet transform (SWT) preserves spatial alignment across subbands, allowing the network to integrate G-buffer attributes and temporally warped history frames in a shift-invariant manner. The predicted coefficients are recombined through inverse wavelet synthesis, producing resolution-consistent reconstructions across arbitrary scale factors. We conduct extensive evaluations and ablations, showing that incorporating SWT yields superior perceptual quality compared to industry baselines, while maintaining real-time performance on modern hardware. Taken together, our results suggest that wavelet-domain neural super-resolution provides a principled and efficient path toward higher-quality real-time rendering, with broader implications for neural rendering and graphics applications.

1. Introduction

The demand for high-resolution graphics at real-time frame rates continues to push modern rendering pipelines to their computational limits. Physically based shading and ray tracing greatly improve realism but are often too expensive to run at native resolutions. Image-space super-resolution has therefore become an essential component of real-time graphics systems: frames are rendered at lower resolution and then reconstructed to display resolution, trading computation for learned upscaling.

Neural approaches have shown strong potential in this space. Methods such as NSRR [12], FuseSR [16] and DFASR [14] combine low-resolution shading with auxiliary scene information such as G-buffers and motion vectors to reconstruct temporally coherent images. We adopt DFASR as a representative rendering-aware baseline because it supports arbitrary-scale reconstruction and exposes a modular pipeline, enabling controlled changes to the prediction target without altering the overall system design.

Despite these architectural advances, existing methods typically regress final pixel values directly in the spatial RGB or irradiance domain. This direct spatial regression forces the network to simultaneously predict broad, low-frequency lighting gradients and sharp, high-frequency structural details within the same output space. The resulting spectral mixing often forces the network to compromise, leading to over-smoothed textures, loss of fine specular highlights, or residual blurring in highly detailed regions.

To address this limitation, we propose a frequency-aware wavelet-domain formulation for neural super-resolution tailored to real-time rendering pipelines. Rather than directly predicting RGB intensities, our model regresses wavelet coefficients that explicitly separate and encode localized multi-scale detail, improving high-frequency recovery and reducing blur in textured regions. We employ the stationary wavelet transform (SWT) and embed inverse wavelet synthesis into the network for end-to-end training. The resulting architecture retains arbitrary-scale upscaling while reformulating the output space to wavelet coefficients for reconstruction. While absolute performance depends on hardware, our formulation provides a practical path toward higher-fidelity reconstruction within real-time rendering constraints.

2. Related Work

This section surveys prior work most relevant to our setting. We begin with neural super-resolution for real-time rendering, focusing on methods that leverage engine metadata, high-resolution G-buffers, and coordinate-based predictors. We then review wavelet-based super-resolution, contrasting discrete and stationary transforms and common strategies that supervise in coefficient and image space. Finally, we summarize Bidirectional Reflectance Distribution Function (BRDF) demodulation and evaluation protocols, including perceptual metrics, which frame our experimental design and analysis.

⋆

*Corresponding author
prateekpoudel61@gmail.com (P. Poudel); aryalprashant07@gmail.com (P. Aryal); kunwarkirtan36@gmail.com (K. Kunwar); navinkhanal3@gmail.com (N. Nepal); dinesh@ioe.edu.np (D.B. Kshatri)

ORCID(s): 0009-0006-1947-825X (P. Poudel); 0009-0004-6756-6807 (P. Aryal); 0009-0001-7913-1497 (K. Kunwar); 0009-0007-8733-3826 (N. Nepal)

https://www.linkedin.com/profile/view?id=prateekpoudel (P. Poudel), https://www.linkedin.com/profile/view?id=pr-a-sh-ant (P. Aryal), https://www.linkedin.com/profile/view?id=kirtan-kunwar (K. Kunwar), https://www.linkedin.com/profile/view?id=navin-nepal (N. Nepal)

[1]These authors contributed equally to this work.

2.1. Neural Super-Resolution for Rendering

Early neural approaches to real-time upsampling combine low-resolution (LR) shading with engine metadata (e.g., depth and motion vectors) and temporal models to reconstruct high-resolution (HR) frames. NSRR [12] exemplifies this direction, achieving temporally coherent upsampling using dense motion and depth cues, but such methods remain constrained by the information content of the LR input and typically operate at fixed scale factors.

In parallel, industry systems such as NVIDIA Deep Learning Super Sampling (DLSS) [6] integrate neural reconstruction into the rendering pipeline by leveraging temporal history and engine-provided signals to produce display-resolution output. Later research further incorporates high-resolution G-buffers to inject geometry and materialaware detail efficiently; FuseSR [16] uses multi-resolution fusion to align LR color with HR buffers for temporally consistent 4× and 8× upsampling, though it remains tied to discrete scales and accurate multi-resolution alignment. ExtraNet [3] instead targets latency by extrapolating future frames using motion cues and HR buffers.

DFASR [14] advances arbitrary-scale reconstruction via a Fourier-mapped implicit representation that predicts pixel values at requested coordinates while exploiting auxiliary modules for robustness. We adopt DFASR as the reference framework because it is a representative rendering-aware architecture that supports arbitrary-scale reconstruction and explicitly incorporates high-resolution G-buffers and temporal information. Its modular design allows us to isolate the impact of replacing RGB regression with wavelet-domain coefficient prediction.

Beyond spatial super-resolution, deep learning has also revolutionized reconstruction for real-time ray tracing. [1] proposed recurrent autoencoders for interactive denoising of Monte Carlo sequences, [7] introduced neural radiance caching to accelerate path tracing. Similarly, ReSTIR GI [10] employs resampling techniques to handle sparse lighting signals. While these methods share the broader goal of reconstructing high-fidelity imagery from sparse data, they primarily focus on denoising Monte Carlo samples rather than the arbitrary-scale spatial upscaling of rasterized buffers targeted by our framework.

2.2. Wavelet-based Super-Resolution

Wavelet-based super-resolution represents images using multi-resolution subbands, separating coarse structure from high-frequency detail. Early Convolutional Neural Network (CNN) approaches such as Deep Wavelet Prediction regress wavelet subband coefficients and reconstruct the output via inverse synthesis, demonstrating improved recovery of high-frequency components compared to direct RGB regression [4]. Subsequent works integrate wavelet analysis and synthesis operations into convolutional backbones, reporting consistent gains across image super-resolution benchmarks [13].

Two commonly used transforms are the discrete wavelet transform (DWT) and the stationary wavelet transform (SWT): DWT is computationally efficient, while SWT avoids decimation and produces subbands at the same spatial resolution at higher memory and compute cost [2]. Most wavelet-SR methods use single-level decompositions and may supervise both coefficient and image space to stabilize training, typically using orthogonal families such as Haar (db1) and Daubechies [4, 13]. In parallel, arbitrary-scale reconstruction has also been explored with Fourier-feature implicit neural representations (INRs), which rely on globally supported sinusoidal bases and can be sensitive to phase under small shifts. Wavelet coefficients instead provide a localized multi-scale representation that is well matched to rendered imagery, where high-frequency effects (e.g., specular highlights and shadow boundaries) are spatially confined and closely tied to per-pixel auxiliary signals such as G-buffers and temporally warped history. This motivates exploring wavelet-domain prediction as an alternative output representation for rendering-aware super-resolution.

2.3. BRDFs and Evaluation Metrics

Radiance demodulation is commonly used in rendering-aware super-resolution to decouple illumination from material-dependent reflectance prior to neural reconstruction. Super-resolution is applied to the demodulated lighting signal and the output is re-modulated with high-resolution material information, enabling the network to focus on smoother signals while preserving sharp textures and temporal coherence [5]. Related work employs BRDF-aware formulations and high-resolution G-buffers to stabilize reconstruction under view-dependent effects and improve upsampling quality at higher scales [17, 16]. Rendering-oriented methods such as DFASR combine these ideas with arbitrary-scale reconstruction and temporal masking within the rendering pipeline [14].

For evaluation, most real-time super-resolution methods report PSNR and SSIM, with LPIPS increasingly used to better reflect perceptual quality [15]. As metrics in linear High Dynamic Range (HDR) space are less standardized for real-time rendering, many works including ours evaluate quantitative results and qualitative comparisons on tone-mapped outputs to reflect display-relevant visual quality.

3. Method

Our framework extends recent rendering-aware neural super-resolution models by shifting the prediction domain from RGB pixels to wavelet coefficients. While our architecture follows the general coordinate-based design of DFASR [14], our key distinction lies in formulating super-resolution as a wavelet reconstruction task. By predicting frequency-aware subbands and reconstructing through the stationary wavelet transform (SWT), we preserve spatial resolution across coefficients and improve recovery of fine detail. The following subsections introduce the problem setup, data preprocessing pipeline, and network architecture, culminating in our integration of wavelet-domain reconstruction.

3.1. Overview

Our method formulates real-time super-resolution as a wavelet-domain reconstruction problem. The goal is to recover a high-resolution frame $\hat{I}_{HR}$ from a combination of low-resolution shading, multi-resolution G-buffers, and temporal history. We denote the overall process as

$$\hat{I}_t^{HR} = \mathrm{SR}(I_t^{LR}, G_t^{LR}, G_t^{HR}, I_{t-1}^{HR}, G_{t-1}^{HR}, MV_t, F),$$

where I_t^{LR} is the current low-resolution frame, G_t^{LR} and G_t^{HR} are the low- and high-resolution G-buffers, I_{t-1}^{HR} is the previous high-resolution frame, G_{t-1}^{HR} its corresponding G-buffers, MV_t the motion vectors, and F the material reflectance information. This compact formulation captures the pipeline-level data flow, while the details of each component are discussed in later subsections.

Unlike prior approaches that directly regress RGB values, our network predicts a wavelet representation of the target frame. Specifically, it estimates subband coefficients $\hat{W} = \{\hat{C}_{LL}, \hat{C}_{LH}, \hat{C}_{HL}, \hat{C}_{HH}\}$, which are subsequently combined through an inverse stationary wavelet transform:

$$\hat{I}_t^{HR} = \mathcal{W}^{-1}(\hat{W}).$$

This shift to the frequency domain is particularly well-suited for rendering data: low-frequency lighting and high-frequency material details are naturally separated across subbands, making it easier for the network to reconstruct fine edges, specular highlights, and shading-dependent detail. The result is an upscaling pipeline that aligns more closely with the underlying rendering process while improving fidelity at larger scaling factors.

3.2. Wavelet-Space Formulation

We cast super resolution as prediction in the wavelet domain rather than RGB space. Let $\mathcal{W}_{\tau,b}$ denote a single-level 2D wavelet analysis operator with transform type $\tau \in \{\mathrm{DWT}, \mathrm{SWT}\}$ and orthogonal basis b (for example, Haar, DB4, Sym4). For a target high-resolution image $I_t^{HR} \in \mathbb{R}^{(sH)\times(sW)\times 3}$, the corresponding subbands are

$$W_t \triangleq \{C_{LL}, C_{LH}, C_{HL}, C_{HH}\} = \mathcal{W}_{\tau,b}(I_t^{HR}),$$

where s is the requested scale factor. The network predicts $\widehat{W}_t = \{\hat{C}_{LL}, \hat{C}_{LH}, \hat{C}_{HL}, \hat{C}_{HH}\}$ directly, and the output image is obtained by inverse synthesis,

$$\hat{I}_t^{HR} = \mathcal{W}_{\tau,b}^{-1}\left(\widehat{W}_t\right).$$

We use a single level of decomposition, which balances frequency separation against runtime. Each color channel is decomposed into four subbands, hence twelve coefficient maps in total. For SWT, subbands preserve spatial size, that is $C_* \in \mathbb{R}^{(sH)\times(sW)\times 3}$, which simplifies alignment across branches and avoids resolution loss inside the network. For DWT, subbands are decimated, so we store them in a space-to-depth layout that is convolution friendly and later recover spatial layout with a pixel shuffle prior to inverse synthesis.

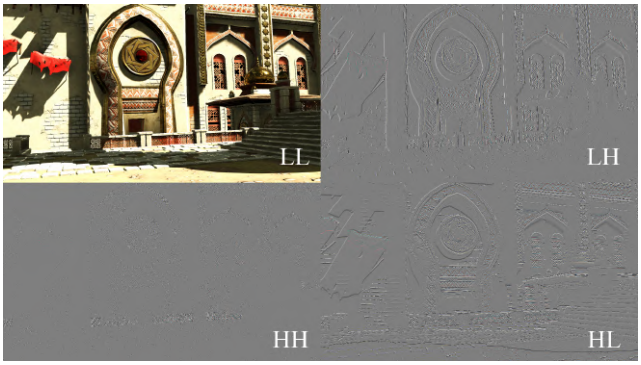

Figure 1: Single-level wavelet decomposition of an RGB image into LL, LH, HH, and HL subbands per channel. Our network predicts these subbands directly, then applies $\mathcal{W}_{\tau,b}^{-1}$ to reconstruct $\hat{I}_t^{HR}$.

The inverse wavelet transform is implemented inside the computational graph so gradients flow through $\mathcal{W}_{\tau,b}^{-1}$ during training.

We employ the stationary wavelet transform (SWT) as the default decomposition in our framework due to its shift-invariant formulation and preservation of spatial resolution across subbands. Unlike decimated transforms, SWT avoids downsampling during analysis, ensuring that all wavelet coefficients are defined on the same spatial grid as the input. This property is particularly important in rendering pipelines, where small spatial shifts arising from reprojection, motion compensation, or G-buffer alignment can otherwise lead to inconsistent coefficient responses across frames and auxiliary signals. The absence of decimation enables more stable coefficient prediction and simplifies fusion with per-pixel metadata at the cost of a modest increase in computation due to non-decimated filtering.

SWT also requires careful boundary handling. To suppress minor edge artifacts from the learned filters and padded convolutions, we crop a one-pixel border when computing losses during training for SWT configurations. The wavelet basis b is treated as a runtime parameter, allowing the use of orthogonal families such as Haar, DB4, or Sym4, while keeping the overall pipeline unchanged. The effect of basis choice and related parameters is analyzed separately in Section 4.4.

3.3. Data Preprocessing and Input Construction

To align super-resolution with the rendering process, our model operates in *irradiance space*, where shaded images are factorized into BRDF and irradiance components. This decomposition allows the network to focus on reconstructing lighting variation separately from view- and material-dependent reflectance. A low-resolution frame I_t^{LR} is demodulated using its per-pixel BRDF term F_t^{LR}:

$$L_t^{LR} = \frac{I_t^{LR}}{F_t^{LR}},$$

Without Clamping | With Clamping | Ground Truth

Figure 2: Visual impact of temporal clamping. In dynamic scenes, disocclusion events can cause the warped history to contain invalid data, leading to ghosting artifacts (Left). By clamping the reprojected history to the neighborhood of the current frame (Middle), we suppress these outliers and restore a clean reconstruction that matches the Ground Truth (Right).

Rendered Frame | BRDF | Irradiance

Figure 3: Illustration of BRDF demodulation. The shaded frame is factorized into BRDF and irradiance. This separation shifts the network's learning focus to low-frequency lighting while high-frequency reflectance patterns are reapplied during remodulation.

where L_t^{LR} denotes the demodulated irradiance. The network then predicts the high-resolution irradiance $\hat{L}_t^{HR}$:

$$\hat{L}_t^{HR} = \mathcal{M}\big(L_t^{LR}, G_t^{HR}, Warp(L_{t-1}^{HR}), \Phi\big),$$

with $\mathcal{M}$ denoting our wavelet-based reconstruction model. Here G_t^{HR} is the high-resolution G-buffers, $Warp(\cdot)$ is temporal reprojection of the previous irradiance frame using motion vectors MV_t. $\Phi = \{\Phi_{\text{spat}}, \Phi_{\text{temp}}\}$ are spatial and temporal masks as introduced in [14].

To mitigate ghosting and accumulation errors during reprojection, we apply a lightweight neighborhood clamping operation. The warped history is constrained to the value range of the local 5×5 neighborhood of the upsampled current frame (expanded by a small tolerance $\epsilon = 0.025$). This ensures that reprojected history remains consistent with the current scene geometry before entering the feature extractor, as demonstrated in Fig. 2.

The G-buffers contain albedo (A), normals (N), depth (D), roughness (R), and metallic (M), from which per-pixel BRDF factors ($F = \{F_t^{LR}, F_t^{HR}, F_{t-1}^{HR}\}$) are derived. After super-resolution, the final shaded output is obtained by remodulation:

$$\hat{I}_t^{HR} = F_t^{HR} \cdot \hat{L}_t^{HR},$$

which restores view- and lighting-dependent effects. This formulation shifts the network's focus toward reconstructing frequency content in irradiance space, while high-resolution geometric cues from G_t ensure accurate remodulation in the final image.

3.4. Network Architecture

Our model adapts the DFASR backbone but shifts the output space to wavelet coefficients. As shown in Fig. 4, the network processes inputs through dedicated feature extractors, and combines them via a feature fusion path and a Fourier-mapped INR branch. The combined output is reconstructed into the final image using the inverse wavelet transform.

3.4.1. Feature Extractors

Our model employs three dedicated feature extractors tailored to the input modalities: a low-resolution frame extractor (LR-FE), a high-resolution G-buffer extractor (GB-FE), and a temporal extractor (T-FE). Prior to the feature extraction stage, we apply pixel unshuffle (space-to-depth) to the low-resolution, demodulated input. This rearranges samples without interpolation by trading spatial resolution for channel capacity, transforming the tensor from (H, W, C) to $(H/2, W/2, 4C)$.

The **LR-FE** processes the pixel-unshuffled, demodulated low-resolution input frame I_t^{LR}, using a stack of 3×3 convolutions to produce a feature map later routed to both fusion and INR pathways. The **GB-FE** encodes the high-resolution G-buffers G_t^{HR} through Residual-in-Residual (RIR) blocks, extracting geometry- and material-aware information such as depth and surface orientation at display resolution. The **T-FE** receives the warped previous frame $\text{Warp}(I_{t-1}^{HR}, MV_t)$ together with temporal masks, and is implemented with lightweight gated convolutions (LWGC) to balance motion sensitivity and efficiency.

Together, these extractors provide complementary signals: LR features represent coarse appearance, HR G-buffers supply high-frequency geometric guidance, and temporal features capture frame-to-frame coherence. Their outputs serve as the basis for subsequent fusion and wavelet-domain reconstruction. Finally, after coefficient prediction, we apply pixel shuffle (depth-to-space) to restore the spatial layout prior to the inverse wavelet synthesis, providing an efficient resolution recovery step without interpolation.

3.4.2. Feature Fusion

The feature fusion module aggregates information from three complementary sources: low-resolution frame features, high-resolution G-buffer features, and temporally aligned features from the previous frame. This combined representation captures both spatial and geometric context, providing the structural backbone of the reconstruction. The fusion branch is responsible for generating the coarse wavelet-domain prediction, while the implicit neural representation (INR) branch contributes finer local corrections. Their outputs are combined element-wise to form the final

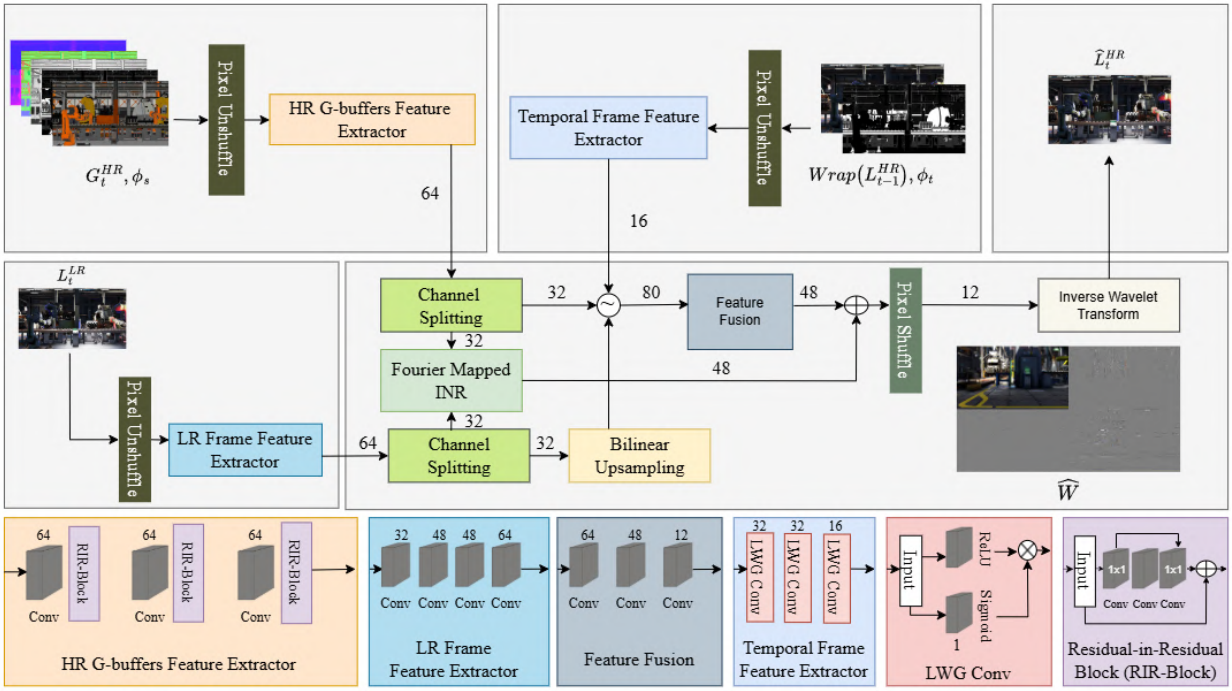

Figure 4: Overview of our network architecture. The model consists of three feature extractors (for LR frames, HR G-buffers, and temporal inputs), a convolutional fusion branch, and an implicit neural representation (INR) branch with Fourier feature mapping. Their outputs are combined in the wavelet domain and reconstructed into the final image via inverse wavelet transform and pixel shuffle. Convolutional blocks are 3×3 with padding 1, except the RIR-blocks which consist of 1×1 expand/reduce layers with an inner 3×3 convolution.

set of predicted wavelet coefficients, which are later synthesized into the high-resolution image.

3.4.3. Implicit Neural Representation (INR)

To support arbitrary-scale super-resolution while enhancing localized high-frequency detail, we employ a coordinate-based Implicit Neural Representation (INR) branch. Prior works such as DFASR [14] demonstrated the effectiveness of Fourier-mapped INRs for rendering-aware upscaling. Building on this idea, we adapt the design to predict wavelet-domain features rather than RGB pixels, which provides finer frequency separation and improved detail recovery.

For each HR pixel coordinate x, we compute a Fourier embedding

$$\mathcal{F}(x) = A(x) \cdot \begin{bmatrix} \cos(\pi \langle F(x), x \rangle + \phi(x)) \\ \sin(\pi \langle F(x), x \rangle + \phi(x)) \end{bmatrix},$$

where $A(x)$ is an amplitude term derived from LR features, $F(x)$ is a frequency term guided by HR G-buffer features, and $\phi(x)$ is a spatial phase offset. These embeddings are passed through a lightweight multi-layer perceptron (MLP), producing predictions in wavelet space.

The INR complements the fusion module by injecting coordinate-aware, high-frequency corrections. Their outputs are combined element-wise to form the final set of predicted coefficients. Together, the two branches balance structural fidelity with detail sharpness, while the continuous INR mapping enables reconstructions at arbitrary scale factors.

3.4.4. Reconstruction

The final reconstruction stage aggregates the outputs from the feature fusion and INR branches, which are element-wise combined to form the estimated wavelet coefficients. These coefficients are subsequently passed through a reconstruction layer consisting of a lightweight convolution and a single-level inverse wavelet transform (IWT), following the formulation already introduced in Sec. 3.2. This yields the super-resolved output image.

Our implementation supports configurable reconstruction settings: the wavelet basis can be selected at runtime (e.g., Haar, Daubechies), and both discrete (DWT) and stationary (SWT) variants are available. By default, the reconstruction is performed with a single-level discrete transform, but the flexibility of our framework allows controlled experimentation with different bases and transform types.

The resulting output is produced in HDR linear space. For visualization purposes, we apply tonemapping during evaluation, but training and inference operate directly in the HDR domain to preserve photometric consistency.

3.4.5. Loss Functions

Training is guided by a composite objective that balances fidelity in both the image and wavelet domains. We combine (i) an ℓ_1 loss on predicted wavelet coefficients, (ii) an ℓ_1 reconstruction loss in image space, (iii) perceptual similarity terms using SSIM and LPIPS, and (iv) spatial mask and temporal consistency losses following DFASR [14]. The

total loss is

$$\mathcal{L}_{total} = \lambda_w \mathcal{L}_{waveletL1} + \lambda_i \mathcal{L}_{imageL1} + \lambda_s \mathcal{L}_{SSIM} + \lambda_p \mathcal{L}_{LPIPS} + \lambda_m \mathcal{L}_{mask} + \lambda_t \mathcal{L}_{temporal}$$

where the weighting coefficients are set to $\lambda_w = 0.1$ for wavelet consistency, $\lambda_i = 1.0$ for spatial reconstruction, $\lambda_s = 0.4$ for structural similarity, $\lambda_p = 0.2$ for perceptual alignment (LPIPS), $\lambda_m = 0.1$ for mask regularization, and $\lambda_t = 0.3$ to enforce temporal coherence. This formulation enforces accuracy across spectral, perceptual, and temporal dimensions, ensuring stable reconstructions under real-time rendering conditions.

4. Results

4.1. Evaluation Setup

We evaluate our method on a custom Unreal Engine 4.27 dataset comprising diverse indoor and outdoor content, including urban environments, industrial interiors, and stylized game assets. The dataset contains approximately 3700 training frames, 700 validation frames, and 1000 test frames, sampled from seven distinct scenes (Table 1). For each frame, we render a high-resolution (HR) target at 1920×1080 together with the corresponding HR G-buffers and auxiliary signals required by our pipeline. Low-resolution (LR) inputs are generated by nearest-neighbor downsampling at random scales between 2.0× and 4.0× during training, enabling arbitrary-scale inference at test time. Data extraction is performed using a custom Unreal Engine plugin adapted and extended from the ExtraNet engine modification [3], which automates synchronized capture of shading outputs and auxiliary buffers across scenes and sequences. To support reproducibility, we publicly release both the implementation (including the capture plugin) and the captured dataset (scene sequences and exported buffers), enabling reproduction of the results reported in this paper and facilitating future research.

The dataset spans a wide range of visual characteristics, covering both stylized environments (e.g., Eastern Village, Brass City) and heavily photorealistic settings (e.g., Abandoned Factory, City Park, Factory Interior, Slay, Downtown West). These assets contain heterogeneous materials and varying degrees of geometric complexity. For instance, while Downtown Alley features relatively simple textures and planar geometry, scenes like City Park and Downtown West introduce substantial high-frequency structural details through dense foliage and thin props. Lighting also varies from strong outdoor sunlight with sharp shadows to mixed indoor or artificial illumination with heavy occlusion. Furthermore, environments such as Factory Interior and Brass City were explicitly chosen to test the reconstruction of complex metallic surfaces and view-dependent specular highlights. This overall diversity is intended to rigorously evaluate the network's robustness across both scene composition and lighting complexity.

Table 1
Dataset composition with train/validation/test split for each scene. Training data are collected from short video sequences (185 frames per sequence) to support temporal evaluation. All frames are rendered at 1920×1080 with HR G-buffers and downsampled to form LR inputs.

Scene	Train	Val	Test
Eastern Village	555	100	285
Abandoned Factory	740	100	0
Brass City	740	100	0
City Park	740	100	0
Factory Interior	555	100	185
Slay	370	100	185
Downtown Alley	0	100	185
Downtown West	0	0	285
Total	3700	700	1000

4.2. Quantitative Comparison

We compare our proposed wavelet-space super-resolution framework against three representative baselines: (i) classical interpolation (Bicubic), (ii) general-purpose image SR (Real-ESRGAN [11]), and (iii) rendering-oriented neural SR (DFASR [14]). Evaluation is conducted at 3× upscaling, reporting PSNR, SSIM, and LPIPS, as summarized in Table 2.

Bicubic serves as a simple baseline but produces heavily smoothed images that lack high-frequency recovery. Real-ESRGAN produces sharper outputs than Bicubic, but its natural-image prior is not optimized for rendered content and can introduce texture details that are inconsistent with scene geometry. In contrast, rendering-aware approaches (DFASR and our method) leverage G-buffers and BRDF demodulation to produce sharper and more robust reconstructions.

Table 3 reports the performance of different wavelet configurations at 3× upscaling, averaged across test scenes. Our best-performing variant, **SWT with DB4 wavelet**, achieves the highest PSNR and SSIM and the lowest LPIPS among the tested configurations. Compared to the "No Wavelet" baseline, it improves PSNR by **1.5 dB** and reduces LPIPS by an **absolute 0.022** on average. While SWT–DB4 incurs slightly higher runtime cost than DWT variants (see Section 4.6), it provides the best overall trade-off between reconstruction quality and efficiency in our experiments.

We report Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS) [15]. PSNR and SSIM are computed in normalized HDR space with gamma correction and clipping. LPIPS is computed using AlexNet for training monitoring and VGG for final reporting. Metrics are averaged across scales of 2×, 3×, and 4× for each test scene. At 4× upscaling, absolute PSNR and SSIM decrease across all methods, reflecting the difficulty of extreme magnification. Table 4 details the performance at this aggressive scaling factor. While numerical fidelity drops, our SWT-based model maintains lower LPIPS than DFASR. Figure 6 visually corroborates this stability, showing that our method

Table 2
Comparison with Bicubic, Real-ESRGAN, and DFASR on test scenes (3× upscaling). Best results in bold. Our SWT DB4 configuration consistently achieves the best performance.

Method	Eastern Village			Downtown Alley			Downtown West		
	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑
Bicubic	23.9	0.352	0.760	30.5	0.326	0.837	23.5	0.420	0.678
Real-ESRGAN	24.7	0.305	0.760	30.1	0.251	0.825	24.4	0.326	0.678
DFASR	28.2	0.180	0.850	32.5	0.152	0.903	26.7	0.250	0.763
SWT DB4 (Ours)	**29.7**	**0.115**	**0.888**	**36.8**	**0.059**	**0.960**	**28.5**	**0.144**	**0.872**

Table 3
Quantitative comparison of methods (3x upscaling). Best results in bold.

Method	PSNR↑	LPIPS↓	SSIM↑
No Wavelet	28.9	0.136	0.897
DWT Haar	29.4	0.124	0.905
SWT Haar	29.7	0.117	0.912
SWT DB4	**30.4**	**0.114**	**0.920**

Table 4
Quantitative results on the Slay scene for **2× and 4× upscaling**. Comparing the degradation from 2× to 4×, our method exhibits a smaller relative drop in PSNR and SSIM than DFASR, indicating improved stability at higher scaling factors.

Configuration	PSNR ↑	LPIPS ↓	SSIM ↑
DFASR 2x	29.25	0.211	0.875
DFASR 4x	27.90	0.249	0.852
Our 2x	32.05	0.181	0.904
Our 4x	31.43	0.192	0.895

avoids the structural collapse often seen in baselines at low input resolutions.

Furthermore, as a preliminary quantitative assessment of temporal stability, we evaluated the Video Multi-Method Assessment Fusion (VMAF)[8] metric on select scenes. Our method achieves scores of 97.23 on the Abandoned Factory sequence and 78.80 on the Eastern Village sequence, indicating reasonable temporal coherence. While these initial results are encouraging, a more comprehensive analysis of temporal stability under highly complex motion profiles and varying frame rates remains necessary to fully characterize long-term performance.

All models are trained end-to-end with a patch size of 432×432 and batch size of 8. Evaluations are performed on full-resolution frames to ensure fidelity across the entire image.

4.3. Qualitative Comparison

While quantitative metrics such as PSNR, SSIM, and LPIPS provide useful averages, they do not always capture perceptual differences. We therefore present visual comparisons against key baselines, focusing on regions with fine textures, sharp edges, and high-frequency structures.

Classical interpolation (Bicubic) produces overly smoothed reconstructions with little recovery of fine detail, while Real-ESRGAN [11] generates sharper results but still leaves high-frequency textures blurry and occasionally hallucinates patterns inconsistent with geometry. Rendering-aware methods show clearer advantages: the no-wavelet baseline (NoWave) and DFASR preserves overall structure and decent sharpness but often introduces ringing or residual blur. Our SWT-based model provides modest but consistent improvements in edge clarity and texture fidelity, particularly in repeated patterns, foliage, and diagonal structures. As shown in Figure 5, SWT restores edges more faithfully than both DFASR and NoWave, and in several scenes demonstrates improved color fidelity and reduced errors, resulting in more visually consistent reconstruction.

These visual observations align with the quantitative trends, indicating that the wavelet-space formulation provides a small but reliable improvement in high-frequency detail reconstruction.

4.4. Ablation Study

To better understand the contributions of different design choices, we group ablations into two categories: *wavelet formulation* and *architectural/training variants*. All results are reported at 2× upscaling on tone-mapped outputs.

Wavelet formulation. We compare discrete wavelet transform (DWT) against stationary wavelet transform (SWT), and evaluate different wavelet bases. This isolates the effect of transform type and basis on reconstruction quality.

Multi-level decomposition. To investigate whether deeper frequency analysis yields better reconstruction, we evaluated a 2-level stationary wavelet transform (SWT2L) using the Haar basis. As shown in Table 6, the 2-level variant achieves a PSNR of 30.2 dB, offering a modest gain of 0.3 dB over the single-level SWT-Haar (29.9 dB). However, this improvement comes with a disproportionate computational penalty. As detailed in Section 4.6, the redundancy of the SWT means that adding a second level significantly expands the memory footprint and compute load. Since the primary goal of our framework is real-time rendering, the substantial

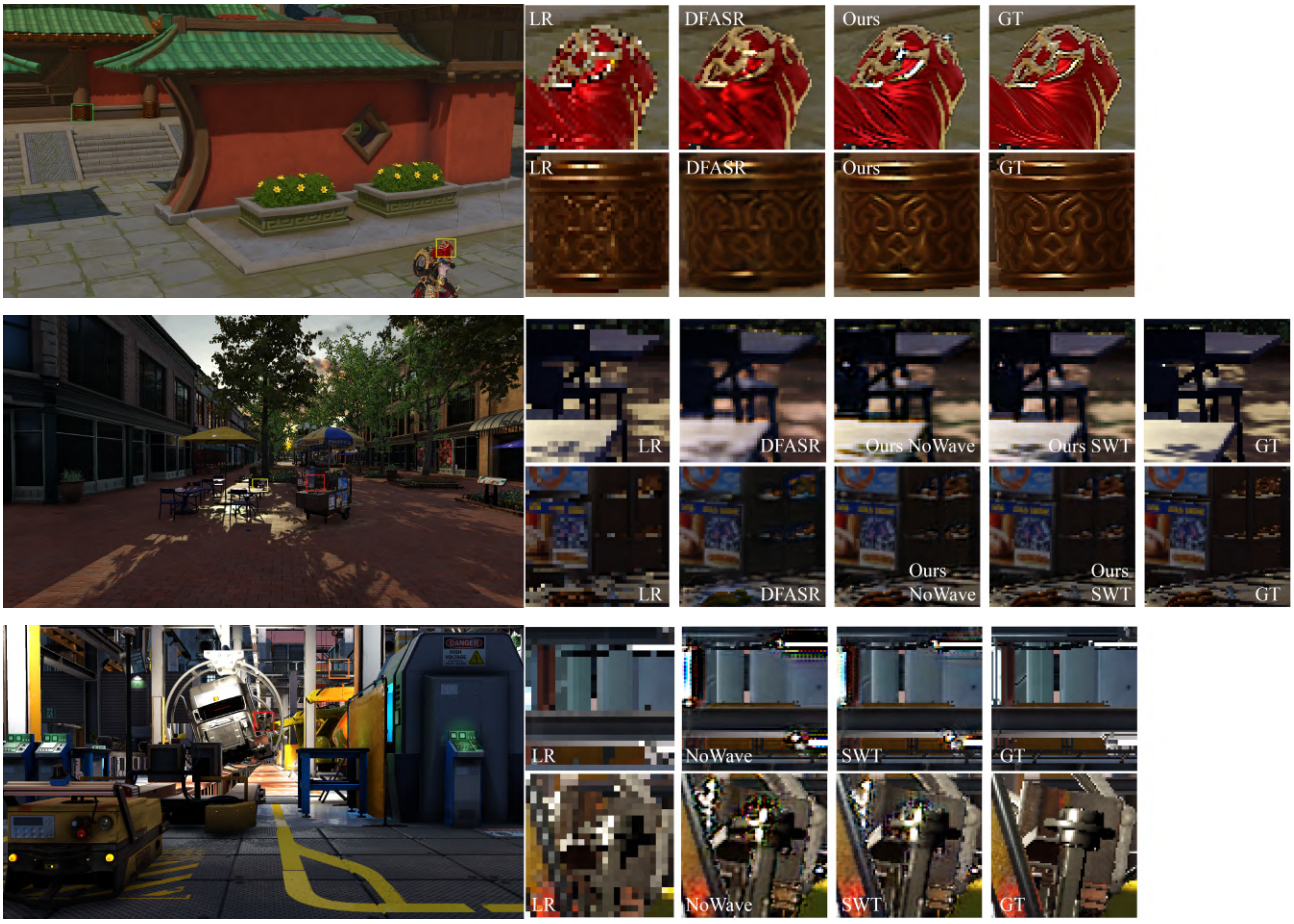

Figure 5: Upscaling results on multiple scenes using NoWave, DFASR [14], and our SWT model. The first scene (Eastern Village) demonstrates that our SWT model recovers sharper details on the head region, where specular highlights are preserved more faithfully compared to DFASR. In the second scene (Downtown West), our method produces clearer structures on the pillars and achieves more accurate color reconstruction, while also maintaining sharper edges in fine elements such as tables. The third scene (Factory) highlights the advantage of SWT in handling challenging color regions, where our model more reliably reconstructs white surfaces that NoWave tends to degrade. Overall, SWT consistently preserves edges and fine details better than the baselines, producing sharper, cleaner, and more realistic outputs across both 2× (first two scenes) and 3× (third scene) upscaling.

Figure 6: Visual consistency of our method across 2×, 3×, and 4× upscaling factors compared to the Low Resolution (LR) input and Ground Truth (GT) on the Slay scene. The top row highlights the reconstruction of shadowed geometry (foot), while the bottom row demonstrates the recovery of high-frequency specular details (shoulder armor). While a natural reduction in fine detail is observed at 4× due to the limited input information, our method maintains structural coherence and correct geometry, avoiding the severe artifacts or shape collapse typically seen at high scaling factors.

latency increase of multi-level decomposition outweighs the marginal quality gains, leading us to adopt the single-level formulation for our final model.

Architectural and training variants. We further investigate alternative network designs: (1) **Fusion→LL, INR→others (SWT–Haar)** restricts the fusion path to predict the low-frequency LL subband, while the INR predicts the high-frequency subbands (LH, HL, HH). (2) **Multi-INR (DWT–Haar)** assigns a separate INR branch to each subband instead of a shared INR, testing whether coefficient-wise

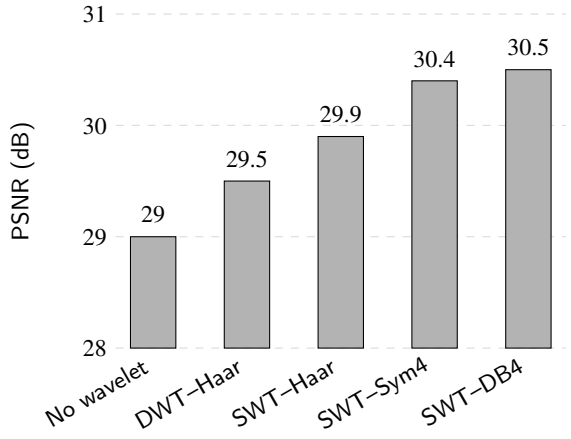

Figure 7: PSNR comparison of ablation configurations at 2× upscaling. Values correspond to Table 5. We plot PSNR only for clarity, as SSIM and LPIPS follow the same trend.

Table 5
Wavelet formulation ablation at 2× upscaling. Metrics are computed on tone-mapped outputs. Best results are highlighted in bold.

Configuration	PSNR ↑	LPIPS ↓	SSIM ↑
No wavelet (baseline)	29.0	0.135	0.899
DWT–Haar	29.5	0.122	0.906
SWT–Haar	29.9	0.114	0.913
SWT–Sym4	30.4	0.116	0.921
SWT–DB4	**30.5**	**0.112**	**0.922**

Table 6
Architectural and training ablations at 2× upscaling. Each variant is evaluated with its native wavelet transform.

Configuration	PSNR ↑	LPIPS ↓	SSIM ↑
SWT-Haar	29.9	0.114	0.913
SWT2L-Haar	30.2	0.114	0.919
Fusion-LL (SWT-Haar)	29.7	0.120	0.910
DWT-Haar	29.5	0.122	0.906
Multi-INR (DWT–Haar)	29.4	0.126	0.905
Single-scale (DWT–Haar)	29.7	0.123	0.909

specialization improves results. (3) **Single-scale (2× only, DWT–Haar)** trains the model exclusively on 2× upscaling instead of arbitrary-scale sampling, measuring whether multi-scale training compromises single-scale fidelity.

Takeaways. Overall, the ablation results show that SWT consistently outperforms DWT, with the DB4 basis giving the highest reconstruction quality among all tested configurations. Alternative architectural variants such as Fusion-LL and Multi-INR reduce performance, indicating that a unified INR with joint coefficient prediction is more effective. Finally, multi-scale training improves generalization without sacrificing fixed-scale accuracy, demonstrating the robustness of our chosen setup.

Table 7
Comparison with NVIDIA DLSS 3.5 (2× Upscaling). Our SWT-DB4 method (2× Upscaling) achieves superior structural and perceptual metrics on both seen (Factory) and unseen (Downtown West) environments.

Method	PSNR ↑	LPIPS ↓	SSIM ↑
Seen Environment: Factory			
NVIDIA DLSS 3.5	26.06	0.274	0.841
Ours (SWT-DB4)	**26.48**	**0.122**	**0.920**
Unseen Environment: Downtown West			
NVIDIA DLSS 3.5	24.20	0.294	0.718
Ours (SWT-DB4)	**28.60**	**0.142**	**0.874**

4.5. Comparison with Industry Standard (DLSS)

To benchmark our framework against a widely used industry standard, we performed a qualitative and quantitative comparison with NVIDIA DLSS 3.5 [9]. To establish a baseline and set clear visual expectations, both DLSS (Performance Mode) and our SWT-DB4 model were evaluated at exactly 2× upscaling. We assessed the models on two distinct scenarios: the *Factory* scene (an unseen camera trajectory within a trained environment) and the *Downtown West* scene (an entirely unseen dataset excluded from training). We emphasize that this is not a rigorously controlled comparison; due to the closed-source nature of the DLSS pipeline, frame captures are not pixel-exact and utilize different tone-mapping operators. Furthermore, DLSS is a highly optimized production system designed to operate under extremely strict hardware and latency constraints. Nevertheless, the local texture content and viewing angles are sufficiently comparable to allow for a general assessment of algorithmic reconstruction characteristics.

As summarized in Table 7, our method demonstrates highly competitive reconstruction quality across both scenes. On the *Factory* environment, our method achieves comparable PSNR (∼26.5 dB vs. 26.1 dB) while offering a noticeable advantage in perceptual quality (LPIPS 0.122 vs. 0.274). On the completely unseen *Downtown West* dataset, our model exhibits strong robustness to novel data, yielding a higher PSNR (+4.4 dB) and lower perceptual error (LPIPS 0.142 vs. 0.294). This suggests that our frequency-aware formulation degrades gracefully and maintains structural integrity even when evaluated far outside of its specific training distribution.

Visual results in Figure 8 corroborate these quantitative trends. While DLSS consistently provides a temporally stable and clean image, it occasionally exhibits a tendency to over-smooth high-frequency textures and simplify thin geometry. In contrast, our wavelet-domain formulation manages to preserve significantly more surface detail, granularity, and structural coherence across both the stylized metals of the *Factory* (Right) and the complex organic structures of novel environments.

Figure 8: Visual comparison between NVIDIA DLSS 3.5 and our SWT-based method (both operating at 2× upscaling). **Left (Downtown West):** On a completely unseen dataset, DLSS struggles with high-frequency structures, washing out the metal fencing, foliage, and distant text. Our method successfully preserves these thin details and textures. **Right (Factory):** On an unseen trajectory within a trained environment, our method retains superior sharpness in the specular highlights of the metal and the diagram on the control panel. **Note:** Due to the closed-source nature of the DLSS debug pipeline, the frames are captured from similar but not pixel-exact timestamps. Slight color variations arise from differences in the respective tone-mapping pipelines.

4.6. Runtime and Efficiency

We evaluate inference performance at 2× upscaling for 1920 × 1080 output on an RTX 3050 mobile GPU, using single-frame inference without low-level optimizations. Figure 9 reports average per-frame runtimes across methods. Our No-Wavelet baseline is close to DFASR, with a minor discrepancy of ∼6 ms, which we attribute to implementation-level differences. Introducing wavelet transforms increases runtime: DWT variants add ∼5 ms overhead relative to the No-Wavelet baseline, while SWT adds ∼18 ms. We also observe that a 2-level SWT (SWT2L) further increases runtime to ∼163 ms, a significant cost increase that yields diminishing returns in quality. This reflects a clear accuracy–efficiency trade-off, where the single-level SWT with DB4 provides the optimal balance.

Table 8 details the cost distribution. The explicit Inverse Wavelet Transform (IWT) step consumes ∼7.1 ms. The remaining portion of the performance overhead (compared to the No-Wavelet baseline) stems from the increased dimensionality of the prediction task: unlike standard RGB regression, the SWT formulation requires the network to predict a spatially redundant set of coefficients (4× the data points of the spatial domain). Despite this increased throughput requirement, the total overhead remains modest relative to the heavy rendering-aware backbone, confirming that the cost of modeling high-frequency wavelet coefficients is manageable on modern hardware.

To demonstrate feasibility for real-time rendering, we evaluated the model on an NVIDIA RTX 3090. Our method achieves an inference time of 13.54 ms (∼73 FPS). This is directly comparable to the DFASR baseline, which reports ∼11.27 ms on similar hardware [14]. The minor increase of ∼2.3 ms is an acceptable trade-off for the gains in perceptual quality (LPIPS/SSIM) demonstrated in Section 4.2. Note that this measurement represents isolated network latency, not total frame time. Achieving strict real-time performance (e.g., 60 FPS) requires the combined base rendering time and inference overhead to remain within a 16.6 ms budget. Consequently, reaching these interactive framerates currently necessitates high-end hardware or scenes with lightweight rendering costs.

5. Conclusion

We have presented a wavelet-domain super-resolution framework for real-time rendering pipelines, predicting frequency-aware subbands rather than directly regressing RGB values. By combining convolutional feature extractors, an implicit neural representation with Fourier mapping, and stationary wavelet reconstruction, our method achieves detail-preserving, resolution-consistent upscaling across arbitrary scales. Experimental results show that this formulation effectively recovers high-frequency detail while remaining compatible with standard rendering pipelines.

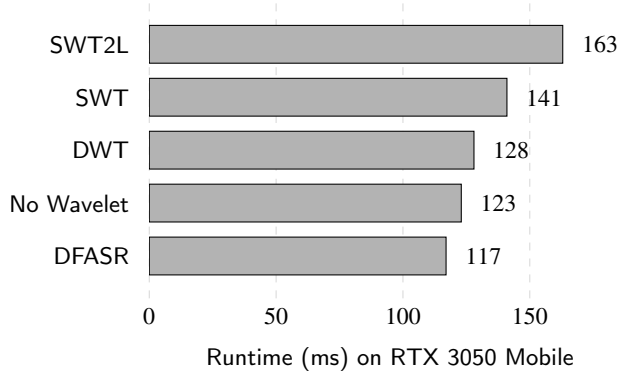

Figure 9: Runtime comparison on RTX 3050 ($720p \rightarrow 1080p$). Our SWT model adds modest overhead (~18 ms) over the No-Wavelet baseline, primarily due to the larger channel depth in the last layer and the IWT step.

Table 8
Component-wise runtime breakdown on RTX 3050 (Total: ~141 ms). The Inverse Wavelet Transform (IWT) accounts for a small fraction of the total inference time.

Module	Time (ms)
LR Feature Extractor	4.15
HR G-Buffer Extractor	34.82
Temporal Extractor	13.15
Feature Fusion	39.46
FMINR	39.41
Inverse Wavelet (IWT)	**7.13**
Other (Upsampling, etc.)	3.5

Several challenges remain. While our temporal clamping improves stability, achieving the long-term robustness of fully recurrent video SR methods remains an area for improvement. Additionally, further optimization is required to ensure practical deployment on mid-range hardware, where the computational overhead of network inference consumes a larger portion of the total rendering budget. Future work may address these through stronger temporal modeling and hardware-aware inference strategies. Finally, the closed-source nature of emerging industry-standard models (e.g., proprietary transformer-based architectures) limits direct reproducibility; future research should therefore investigate integrating open transformer backbones into the wavelet domain to exploit richer semantic contexts. Coupling the framework with real-time ray tracing pipelines also offers a promising direction to exploit richer lighting signals within the same frequency-aware formulation.

Acknowledgements

The authors would like to express their sincere gratitude to Er. Saroj Shakya for his invaluable guidance and support as a co-supervisor throughout the duration of this project. The Department of Electronics and Computer Engineering, Thapathali Campus, is gratefully acknowledged for providing the opportunity and resources to carry out this research. The authors also acknowledge Ashim Bhattarai, Ashim Tiwari and Aman Paudel for their assistance with computational resources.

Data Availability Statement

The datasets used in this study consist of synthetic frames generated in Unreal Engine 4.27. These data can be fully reproduced by following the capture pipeline and preprocessing steps provided in our public repository https://github.com/Prateek61/WDSS. The repository includes the Unreal Engine plugin for automated scene capture, as well as scripts for dataset preparation and training. The dataset used in this work is publicly available at https://drive.google.com/drive/folders/1Z8UmfwoLZXW-Dostqlfb88e4iKDHpFhq?usp=sharing. The results reported in the paper can be reproduced using the provided code, data, and instructions.

References

[1] Chaitanya, C.R.A., Kaplanyan, A.S., Schied, C., Salvi, M., Lefohn, A., Nowrouzezahrai, D., Aila, T., 2017. Interactive reconstruction of monte carlo image sequences using a recurrent denoising autoencoder. ACM Trans. Graph. 36. URL: https://doi.org/10.1145/3072959.3073601, doi:10.1145/3072959.3073601.

[2] Deeba, F., Kun, S., Ali Dharejo, F., Zhou, Y., 2020. Wavelet-based enhanced medical image super resolution. IEEE Access 8, 37035–37044. doi:10.1109/ACCESS.2020.2974278.

[3] Guo, J., Fu, X., Lin, L., Ma, H., Guo, Y., Liu, S., Yan, L.Q., 2021. Extranet: Real-time extrapolated rendering for low-latency temporal supersampling. ACM Trans. Graph. 40. doi:10.1145/3478513.3480531.

[4] Guo, T., Mousavi, H.S., Vu, T.H., Monga, V., 2017. Deep wavelet prediction for image super-resolution, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1100–1109. doi:10.1109/CVPRW.2017.148.

[5] Li, J., Chen, Z., Wu, X., Wang, L., Wang, B., Zhang, L., 2024. Neural super-resolution for real-time rendering with radiance demodulation. URL: https://arxiv.org/abs/2308.06699, arXiv:2308.06699.

[6] Liu, E., 2020. Dlss 2.0: Image reconstruction for real-time rendering with deep learning. GPU Technology Conference (GTC). URL: https://developer.nvidia.com/gtc/2020/video/s22698. nVIDIA Technical Presentation.

[7] Müller, T., Rousselle, F., Novák, J., Keller, A., 2021. Real-time neural radiance caching for path tracing. ACM Trans. Graph. 40. URL: https://doi.org/10.1145/3450626.3459812, doi:10.1145/3450626.3459812.

[8] Netflix, 2023. Vmaf: Video multi-method assessment fusion. https://github.com/Netflix/vmaf.

[9] NVIDIA, 2023. Nvidia dlss 3.5: Enhancing ray tracing with ai. https://www.nvidia.com/en-us/geforce/news/nvidia-dlss-3-5-ray-reconstruction/.

[10] Ouyang, Y., Liu, S., Kettunen, M., Pharr, M., Pantaleoni, J., 2021. Restir gi: Path resampling for real-time path tracing. Computer Graphics Forum 40, 17–29. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14378, doi:https://doi.org/10.1111/cgf.14378, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14378.

[11] Wang, X., Xie, L., Dong, C., Shan, Y., . Real-esrgan: Training real-world blind super-resolution with pure synthetic data, in: International Conference on Computer Vision Workshops (ICCVW).

[12] Xiao, L., Nouri, S., Chapman, M., Fix, A., Lanman, D., Kaplanyan, A., 2020. Neural supersampling for real-time rendering. ACM Trans. Graph. 39. URL: https://doi.org/10.1145/3386569.3392376, doi:10.1145/3386569.3392376.

[13] Yang, H., Wang, Y., 2021. An effective and comprehensive image super resolution algorithm combined with a novel convolutional

neural network and wavelet transform. IEEE Access 9, 98790–98799. doi:10.1109/ACCESS.2021.3083577.

[14] Zhang, H., Guo, J., Zhang, J., Qin, H., Feng, Z., Yang, M., Guo, Y., 2024. Deep fourier-based arbitrary-scale super-resolution for real-time rendering, in: ACM SIGGRAPH 2024 Conference Papers, pp. 1–11.

[15] Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O., 2018. The unreasonable effectiveness of deep features as a perceptual metric. URL: https://arxiv.org/abs/1801.03924, arXiv:1801.03924.

[16] Zhong, Z., Zhu, J., Dai, Y., Zheng, C., Chen, G., Huo, Y., Bao, H., Wang, R., 2023. Fusesr: Super resolution for real-time rendering through efficient multi-resolution fusion, in: SIGGRAPH Asia 2023 Conference Papers, pp. 1–10.

[17] Zhuang, T., Shen, P., Wang, B., Liu, L., 2021. Real-time Denoising Using BRDF Pre-integration Factorization. Computer Graphics Forum doi:10.1111/cgf.14411.