Latent Space Super-Resolution for Higher-Resolution Image Generation with Diffusion Models

Supplementary Material

A. Importance of Latent Space Upsampling

One of our key findings is that for reference-based higher-resolution image generation methods [5, 7, 19], quality of output images differs significantly depending on whether the reference is upsampled in RGB space or latent space. We hypothesize that upsampling within the latent space plays a crucial role in preserving the sharpness and detail essential for higher-resolution image generation. In this section, we provide additional qualitative and quantitative experimental results to support our hypothesis.

A.1. Setting

We define several RGB upsampling variants of the existing models DemoFusion [5] and Pixelsmith [19] by modifying their reference upsampling strategies. First, we introduce DemoFusion-rgbBic for the DemoFusion model, where the reference is upsampled in RGB space using bicubic interpolation. For the Pixelsmith model, we define Pixelsmith-rgbLanc, which employs Lanczos interpolation [11] in RGB space for reference upsampling. Pixelsmith-rgbLanc corresponds to the original Pixelsmith model.

Building on these, we further define variants that perform super-resolution (SR) in RGB space using a separate SR network, namely DemoFusion-rgbSR and Pixelsmith-rgbSR. The SR network shares the same architecture and training settings as our LSR module (detailed in Section C.3), with the input and output channels is set to 3 to process RGB images.

In contrast to these variants, LSRNA-DemoFusion and LSRNA-Pixelsmith utilize latent space upsampling through our proposed LSRNA framework. While the original DemoFusion also performs bicubic upsampling in latent space and demonstrates strengths in preserving detail, we instead demonstrate the effectiveness of latent upsampling within the LSRNA framework.

A.2. Analysis

Our qualitative results, presented in Figures C and D for 16× resolution and Figures E and F for 64× resolution, demonstrate consistent trends when comparing RGB upsampling variants to latent upsampling using our LSRNA framework. Specifically, RGB upsampling methods, whether based on interpolation or super-resolution, produce smoother images that lack fine details. In contrast, latent upsampling yields sharper and more detailed results.

Our quantitative results in Table A further confirm these observations that latent upsampling approaches (*i.e.*, LSRNA-DemoFusion and LSRNA-Pixelsmith) consistently outperform their RGB upsampling counterparts. The improvements are particularly evident in patch-based metrics like pFID and pKID, which focus on capturing finer details. These results underscore the critical role of latent space upsampling in enhancing local detail fidelity and textures.

Table A. **RGB vs. Latent Space Upsampling** on OpenImages-Valid (×9). The best results marked in **bold**.

	FID (↓)	KID (↓)	pFID (↓)	pKID (↓)
DemoFusion-rgbBic	134.56	0.0084	37.44	0.0062
DemoFusion-rgbSR	134.55	0.0093	37.35	0.0061
LSRNA-DemoFusion	**132.01**	**0.0053**	**35.95**	**0.0057**
Pixelsmith-rgbLanc	134.31	0.0095	40.64	0.0084
Pixelsmith-rgbSR	134.34	0.0102	44.41	0.0110
LSRNA-Pixelsmith	**132.17**	**0.0077**	**36.71**	**0.0057**

We attribute these findings to the representational characteristics of the latent space. Unlike RGB space, the latent space encodes image features in a compressed form, capturing high-level information. Upsampling within this domain likely leverages these representations to better preserve fine details and sharpness. Conversely, RGB space upsampling is constrained by the raw pixel-level representation, which acts as a bottleneck and hinders the preservation of details and textures. Further exploration is needed to fully understand the underlying reasons behind these differences.

B. Experimental Details

B.1. Comparison

To ensure a fair comparison, all experiments across the main text and appendix are conducted with a consistent setup unless otherwise specified. This includes the use of a fixed random seed, tiled decoding, negative prompts derived from DemoFusion, a guidance scale of 7.5, xFormers [12] enabled with float16 precision, and FreeU [17] disabled. In addition, unnecessary visualization code like intermediate image reconstruction is disabled for runtime measurement.

We employ a DDIM [18] scheduler, with the η parameter set to 0 for reference-based methods, and $\eta = 1$ for non-reference-based methods, as $\eta = 0$ leads to noticeable degradation in quantitative performance for the latter. Existing methods use 50 DDIM steps for higher-resolution generation process, while LSRNA uses 30 steps.

B.2. Patch-Based Metrics

Conventional metrics like FID [8] and KID [1] involve resizing images to 299^2, which can lead to a loss of high-resolution details. To address this issue, inspired by Anyres-GAN [3], we adopt patch-based metrics (pFID and pKID) that focus on local details and textures, which are critical for evaluating high-resolution image generation.

The patch-based metrics are computed by first cropping the original ground truth images to match the aspect ratio of the generated images, followed by resizing them to the same resolution using Lanczos interpolation. Next, 1K-sized patches are cropped from both the generated and ground truth images at 50,000 randomly selected locations. For a fair comparison, a fixed random seed is used to maintain consistency in crop locations both between generated and ground truth images and across different generation methods. These extracted patches are then used to calculate the FID and KID metrics, referred to as pFID and pKID.

C. LSR Training Details

C.1. Data Preparation

To prepare LR-HR latent pairs for training the LSR module, we leverage the real-world dataset [10] to obtain ground-truth HR RGB images. We construct training pairs in two steps: (i) downsampling the HR RGB images to generate LR RGB images, and (ii) encoding the HR and LR RGB images independently using a pretrained encoder $\mathcal{E}$.

Bridging the Domain Gap. To address the domain gap between training and inference, we simply adopt bicubic degradation over complex real-world degradations [20, 21] in step (i). This choice aligns with the inference scenario, where LR images (decoded from LR latents) typically exhibit minimal noise or artifacts. Bicubic degradation avoids the noise or artifacts introduced by real-world degradations while significantly reducing preprocessing time.

In step (ii), directly downsampling HR latents to create LR latents is avoided, as it can cause inconsistencies within the latent manifold. Instead, our approach ensures that LR and HR latents are encoded separately, preserving consistency within their respective manifolds.

Multi-Scale Preparation. To enable multi-scale training for the LSR module, we filter ground-truth HR RGB images with a minimum resolution of 1440 pixels in both height and width. For each HR image, a crop size is randomly selected between $[1056, 1440]$ in multiples of 96 (chosen to align with the downscaling factors and the encoder's compression ratio of 8). The HR image is then divided into non-overlapping patches of the selected crop size, forming HR RGB patches. Each HR patch is subsequently encoded into the latent space.

To create LR counterparts, each HR RGB patch is downscaled by factors of $\times 2$, $\times 3$, and $\times 4$. The resulting LR RGB patches are then encoded using the same encoder to generate LR latent representations. Our data preparation process results in a training dataset comprising a total of 4.7M LR-HR latent pairs with diverse scale.

Table B. **Quantitative comparison of image generation results by LSR training variants** on OpenImages-Valid ($\times 9$). All training is conducted on a single NVIDIA Tesla V100-SXM2 GPU, using SwinIR [13] and RCAN [23] as backbones with LIIF [4] as the upsampler. Based on a balance of training efficiency and performance, we adopt the v1 configuration.

LSRNA-DemoFusion	**v1 (adpoted)**	v2	v3	v4
Params	1.29M	1.29M	1.29M	15.64M
Backbone	SwinIR (Light)	SwinIR (Light)	SwinIR (Light)	RCAN
Initial learning rate	2×10^{-4}	2×10^{-4}	1×10^{-4}	2×10^{-4}
Batch size	32	32	16	32
Training iteration	200K	1000K	200K	200K
Training time	26h	129h	15h	26h
FID (↓)	134.84	134.90	134.28	134.25
KID (↓)	0.0077	0.0077	0.0077	0.0074
pFID (↓)	33.47	33.35	33.75	34.34
pKID (↓)	0.0073	0.0072	0.0074	0.0076

C.2. Batch Construction

Each LR-HR latent pair has varying sizes, necessitating alignment of the spatial dimensions between the input LR latent and target HR latent for batching. During data-loading, we further randomly crop the LR latents to a fixed size of 32×32 pixels. From the corresponding HR latents, 4096 pixels within the cropped region are randomly sampled to serve as the ground truth. Additionally, we avoid data augmentation techniques such as horizontal and vertical flips, as they can lead to deviations in the latent space manifold. Our batching strategy not only ensures efficient training but also enables the LSR to learn mappings from LR to HR latent representations across multiple scales.

C.3. Training

We adopt SwinIR [13] as the backbone for the LSR and LIIF [4] as the upsampler, modifying both input and output channel dimensions to match the latent space dimension of 4. The optimizer used is Adam [9] with an initial learning rate of 2×10^{-4}, scheduled with cosine annealing. Training is performed over 200K iterations with a batch size of 32. The loss function is defined by an L_1 loss in the latent space. We leave the exploration of other loss functions (*e.g.*, perceptual loss [22]) for future work.

Quantitative results of image generation by various LSR training settings are presented in Table B, while also demonstrating the efficiency of the LSR training. Although the LSR is trained on the paired dataset constructed with relatively sparse downscaling factors, it can generalize to arbitrary scaling factors during inference, enabled by our

Table C. **Quantitative comparison of image generation results by Canny edge detection thresholds** on OpenImages-Valid (×9) with DemoFusion. e_{max} is set to 1.2 (default).

lower	upper	FID (↓)	KID (↓)	pFID (↓)	pKID (↓)
0	255	132.01	0.0053	35.95	0.0057
30	180	132.18	0.0055	36.01	0.0057
50	200	132.54	0.0055	36.09	0.0057
60	150	132.19	0.0055	36.12	0.0057

Table D. **Pixel-wise difference based on RNA strength.** Differences are computed after applying RNA compared to no RNA. Histogram matching is applied before computing differences. The scene from the main Figure 7 is used.

e_{min}	e_{max}	Non-edge	Edge	Gap	e_{min}	e_{max}	Non-edge	Edge	Gap
0	0.6	1.36	3.44	**2.09**	0.6	0.6	6.36	9.61	3.25
0	1.2	2.65	6.57	**3.92**	1.2	1.2	25.95	29.06	3.11
0	1.8	4.01	10.27	**6.26**	1.8	1.8	39.12	39.81	0.7

multi-scale training scheme and the generalization capability of LIIF. Our motivation for using LIIF upsampler instead of traditional fixed-scale upsampler [16] lies in its ability to handle arbitrary resolutions with a single LSR module trained once.

D. Edge Detection for RNA

RNA is designed to adaptively add Gaussian noise to specific areas of the upsampled reference latent, focusing on detail-critical regions (*i.e.*, high-frequency regions). Our intuition behind RNA is that introducing irregularities in regions that would otherwise remain flat prompts the diffusion model to synthesize new details in those regions.

To identify these areas, we consider using edge detection algorithms. However, we find that common edge detectors such as Scharr [15], LoG [14], and Gabor [6], which primarily focus on precise object boundaries, tend to produce artifacts such as overgeneration around contours or jagged contours when used as the basis for RNA. We present qualitative results in Figure A using the Scharr edge map, which is known for effectively capturing weak edges. As shown, while strong edges (*e.g.*, on the train's window) are sparsely detected, weak edges (*e.g.*, on the tree) appear with excessively low intensity. This results in artifacts or over-enhanced details around strong edges when RNA is applied.

To address this, we adopt Canny edge detection [2], which allows us to prioritize weak edges by adjusting the lower and upper thresholds. By doing so, we can detect detailed regions rather than strictly connected edge lines, as demonstrated in Figure A. This region-based detection allows RNA to enhance local details effectively without introducing edge-based artifacts.

Table E. **LSR & RNA ablation** on OpenImages-Valid (×9). The best results marked in **bold**.

	FID (↓)	KID (↓)	pFID (↓)	pKID (↓)	Time (sec)
DemoFusion	**131.95**	0.0064	38.75	0.0075	660
LSRNA-DemoFusion (w/o RNA)	132.65	0.0065	37.10	**0.0057**	272
LSRNA-DemoFusion	132.01	**0.0053**	**35.95**	**0.0057**	272
Pixelsmith	134.31	0.0095	40.64	0.0084	289
Pixelsmith-latentBic	142.23	0.0155	67.91	0.0275	291
LSRNA-Pixelsmith (w/o RNA)	137.71	0.0116	46.45	0.0112	181
LSRNA-Pixelsmith	**132.17**	**0.0077**	**36.71**	**0.0057**	182

E. Robustness of RNA

We demonstrate in Table C that generation performance is robust to variations in the Canny's thresholds, as long as they are set to prioritize weak edges. In our implementation, we use a lower threshold of 0 and an upper threshold of 255. We further evaluate the robustness of RNA through additional experiments: Figure B indicates that RNA is quantitatively better than UNA (Uniform Noise Addition), while Table D shows that RNA indeed enhances details where the edge map is activated.

F. Additional Ablation Studies

F.1. Effectiveness of LSR & RNA

We provide additional quantitative results to assess the impact of the LSR and RNA modules on both the DemoFusion and Pixelsmith models. For the original Pixelsmith, which performs upsampling in the RGB space unlike DemoFusion, we introduce an additional variant called *Pixelsmith-latentBic*. This variant replaces the original RGB space upsampling with bicubic interpolation in the latent space.

The results are summarized in Table E. For DemoFusion, incorporating the LSR module enhances performance by providing high-quality latent guidance, improving image generation quality even with fewer denoising steps and without progressive upscaling. The addition of the RNA module further boosts performance by enriching finer details and textures in the generated images.

In case of Pixelsmith, replacing the original RGB upsampling with latent upsampling (*i.e.*, Pixelsmith-latentBic) leads to significant performance degradation. However, applying the LSR module to perform super-resolution in the latent space leads to a noticeable improvement in performance. The RNA module further improves the results and ultimately surpasses the performance of the original model, demonstrating the adaptability and effectiveness of our LSR and RNA modules.

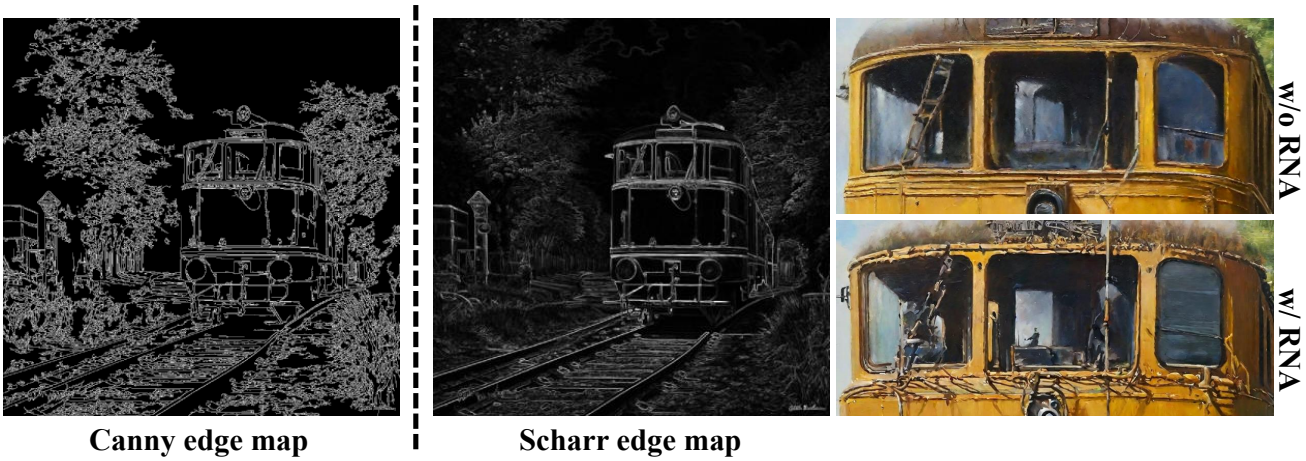

Figure A. **Qualitative results of RNA using Scharr edge map.**

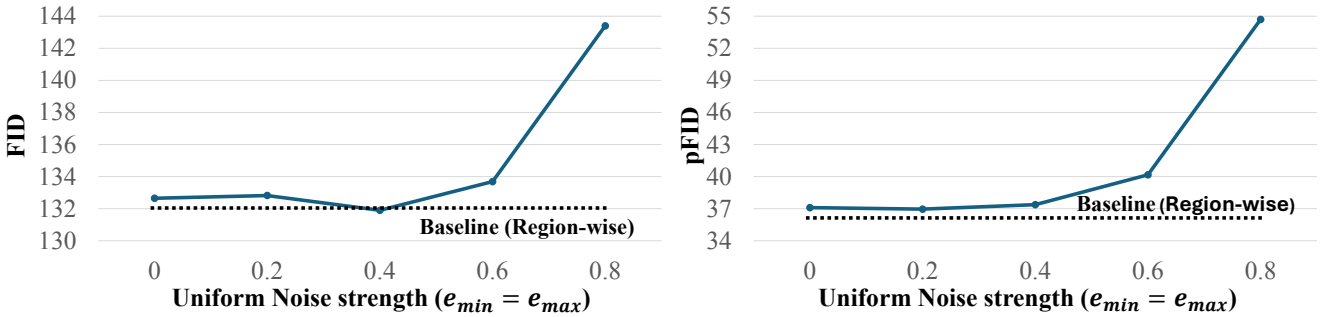

Figure B. **Ablation study of UNA (Uniform Noise Addition) strength** on OpenImages-Valid ($\times 9$) with DemoFusion. Dotted line shows our default RNA setting ($e_{min} = 0$ and $e_{max} = 1.2$).

Table F. **Ablation Study on RNA strength with Pixelsmith** on OpenImages-Valid ($\times 9$). The best results marked in **bold**.

e_{min}	e_{max}	FID (↓)	pFID (↓)	e_{min}	e_{max}	FID (↓)	pFID (↓)
0.0	0.0	137.71	46.45	0.2	1.2	133.67	37.76
0.0	1.2	135.15	40.45	0.2	1.4	133.72	38.51
0.0	1.4	134.32	38.75	0.4	0.6	132.18	37.3
0.0	1.6	134.34	38.74	0.4	0.8	**132.17**	**36.71**
0.2	1.0	133.86	38.51	0.4	1.0	132.29	36.95

F.2. Impact of RNA Strength

Building on the RNA strength tuning results for LSRNA-DemoFusion presented in the main text, we further evaluate the impact of RNA strength on LSRNA-Pixelsmith, as shown in Table F. While LSRNA-DemoFusion achieves optimal performance with $e_{min} = 0$ (and $e_{max} = 1.2$), LSRNA-Pixelsmith performs best with $e_{min} = 0.4$ and $e_{max} = 0.8$. This difference in optimal RNA strength arises from the distinct roles played by the reference latent in the high-resolution generation process of each model. LSRNA-Pixelsmith likely requires a higher e_{min} to ensure effective noise injection into the reference latent. The RNA strength determined from this validation process is consistently applied across all other experiments including those in the main text.

Figure C. **RGB vs. Latent Space Upsampling for DemoFusion on 16×.** Prompt used is ”the sun is setting over the ocean on a cloudy day”. Best viewed **ZOOMED-IN**.

Figure D. **RGB vs. Latent Space Upsampling for Pixelsmith on 16×.** Prompt used is ”the sun is setting over the ocean on a cloudy day”. Best viewed **ZOOMED-IN**.

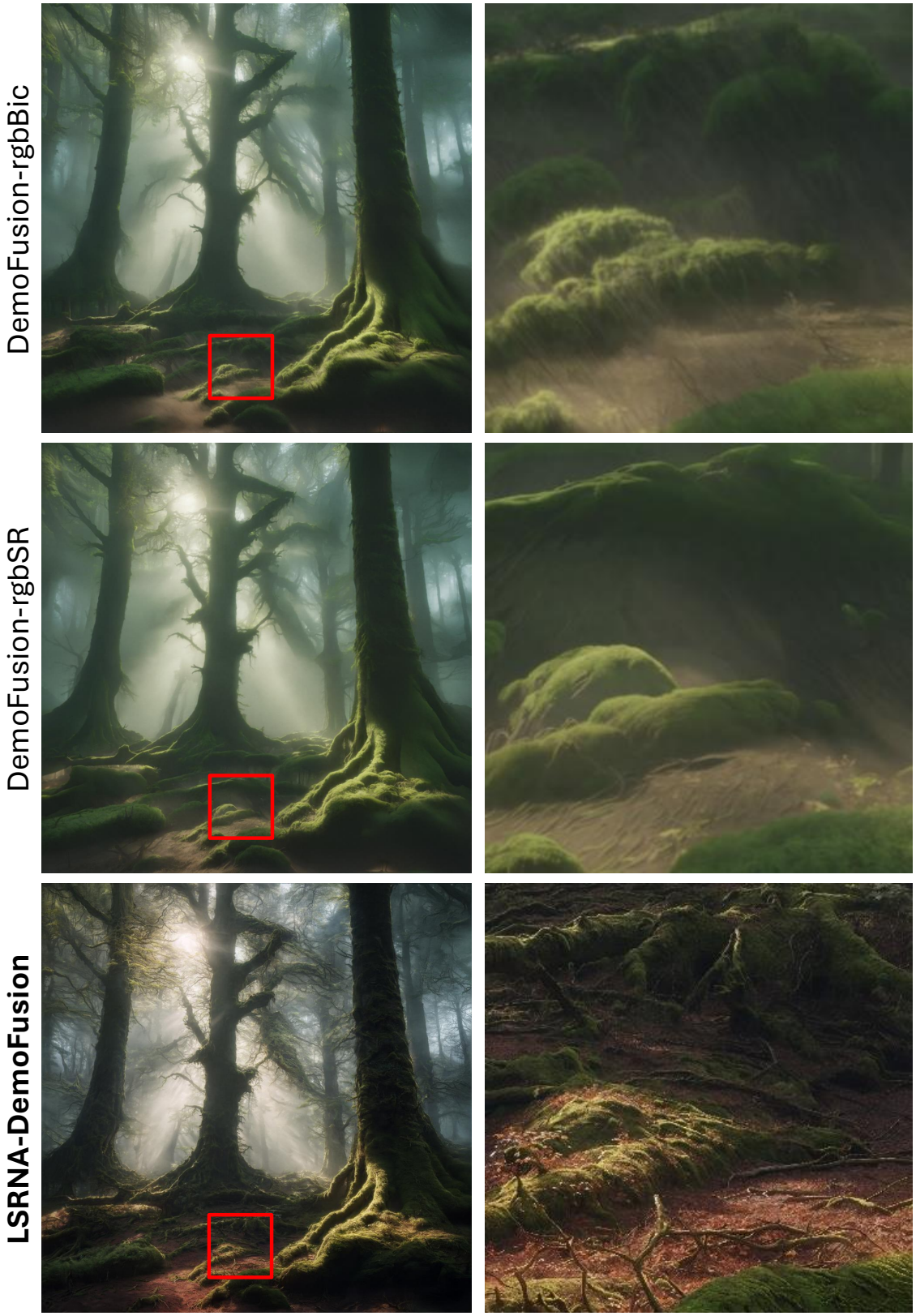

Figure E. **RGB vs. Latent Space Upsampling for DemoFusion on 64×.** Prompt used is ”A mysterious forest with tall, ancient trees and beams of sunlight filtering through the mist, detailed moss-covered roots, 8k”. Best viewed **ZOOMED-IN**.

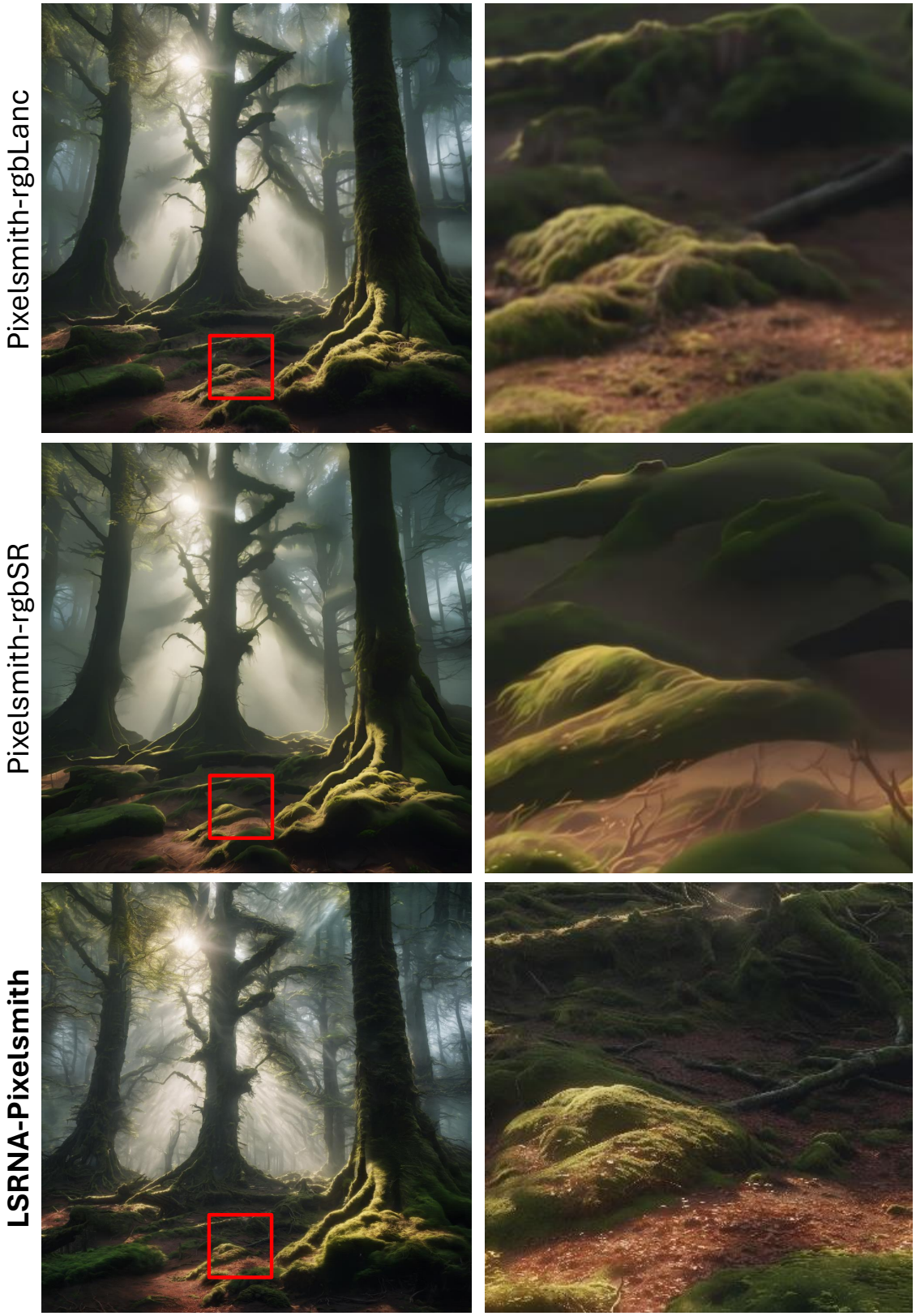

Figure F. **RGB vs. Latent Space Upsampling for Pixelsmith on 64×.** Prompt used is "A mysterious forest with tall, ancient trees and beams of sunlight filtering through the mist, detailed moss-covered roots, 8k". Best viewed **ZOOMED-IN**.

References

[1] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018. 2

[2] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 679–698, 1986. 3

[3] Lucy Chai, Michael Gharbi, Eli Shechtman, Phillip Isola, and Richard Zhang. Any-resolution training for high-resolution image synthesis. In *European Conference on Computer Vision*, 2022. 2

[4] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8628–8638, 2021. 2

[5] Ruoyi Du, Dongliang Chang, Timothy Hospedales, Yi-Zhe Song, and Zhanyu Ma. Demofusion: Democratising high-resolution image generation with no $$$. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6159–6168, 2024. 1

[6] Dennis Gabor. Theory of communication. part 1: The analysis of information. *Journal of the Institution of Electrical Engineers-part III: radio and communication engineering*, 93(26):429–441, 1946. 3

[7] Lanqing Guo, Yingqing He, Haoxin Chen, Menghan Xia, Xiaodong Cun, Yufei Wang, Siyu Huang, Yong Zhang, Xintao Wang, Qifeng Chen, et al. Make a cheap scaling: A self-cascade diffusion model for higher-resolution adaptation. *arXiv preprint arXiv:2402.10491*, 2024. 1

[8] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems*, 30, 2017. 2

[9] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[10] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*, 128(7):1956–1981, 2020. 2

[11] Cornelius Lanczos. Evaluation of noisy data. *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis*, 1(1):76–85, 1964. 1

[12] Benjamin Lefaudeux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, et al. xformers: A modular and hackable transformer modelling library, 2022. 1

[13] 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*, pages 1833–1844, 2021. 2

[14] David Marr and Ellen Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 207(1167):187–217, 1980. 3

[15] Hanno Scharr. Optimal operators in digital image processing. 2000. 3

[16] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016. 3

[17] Chenyang Si, Ziqi Huang, Yuming Jiang, and Ziwei Liu. Freeu: Free lunch in diffusion u-net. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4733–4743, 2024. 1

[18] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 1

[19] Athanasios Tragakis, Marco Aversa, Chaitanya Kaul, Roderick Murray-Smith, and Daniele Faccio. Is one gpu enough? pushing image generation at higher-resolutions with foundation models. *arXiv preprint arXiv:2406.07251*, 2024. 1

[20] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1905–1914, 2021. 2

[21] 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*, pages 4791–4800, 2021. 2

[22] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 2

[23] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *European Conference on Computer Vision*, pages 286–301, 2018. 2