PixelRush: Ultra-Fast, Training-Free High-Resolution Image Generation via One-step Diffusion

Supplementary Material

Overview. The supplementary material provides additional implementation details in Sec. A, discussion on dependence of distilled models in Sec. B, further quantitative results in Sec. C, and an extended gallery of qualitative results in Sec. D.

A. Implementation Details

For the base generation stage, we use the base model SDXL to generate a base image at its native 1024×1024 resolution. We employ a classifier-free guidance scale of 7.5 for all experiments. The reverse process in our refinement stage uses deterministic DDIM sampling, with the DDIM eta parameter set to 0. The upsampling operator $\mathbb{UP}$ is implemented as standard bicubic interpolation performed in the pixel domain (RGB space). Our pipeline operates in a cascaded manner. For example, to generate an 8192×8192 image, the process starts with a 1024×1024 base image, which is then progressively upscaled and refined through 2048×2048 and 4096×4096 resolutions to reach the final target. For the patch extraction process, the latent is tiled into patches with a 50% overlap along both spatial dimensions. For our noise injection technique, the spherical interpolation coefficient λ from Eq. 4 is set to a fixed value of 0.95 for all experiments.

B. Dependence on Distilled Models

We clarify that "training-free" refers to user-side adaptation: no additional training or fine-tuning is required to apply PixelRush. Our reliance on a distilled backbone is a deliberate design choice rather than a limitation, as PixelRush serves as a plug-and-play module compatible with various few-step models (as evidenced in Tab. 3 of the main paper). The potential concern about missing concepts in distilled models is mitigated by our two-stage design: global semantics are established in the coarse stage, while the distilled model is only responsible for refining local structure and texture. Moreover, our noise injection technique actively improves upon distilled models by counteracting their tendency toward texture oversmoothing.

C. Additional Quantitative Results

C.1. Integration with FLUX

To demonstrate the generalizability of PixelRush beyond SDXL, we integrate it with FLUX [8], a state-of-the-art transformer-based diffusion model. As shown in Table 6,

Method	FID (↓)	IS (↑)	Time (sec)
Flux-DI	54.60	15.39	76
Flux+PixelRush	51.73	15.44	20

Table 6. **PixelRush integrated with FLUX** for 2K generation. PixelRush improves both quality and speed over direct FLUX inference.

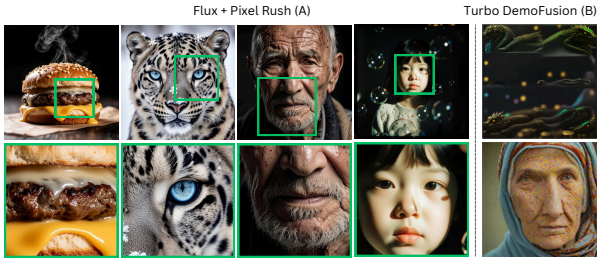

Figure 9. (A) PixelRush integrated with FLUX for 2K generation. (B) Naive "Turbo" DemoFusion fails catastrophically in the few-step regime under both full-noise and partial-noise settings. Best viewed zoomed in.

Flux+PixelRush achieves better perceptual quality than direct Flux inference at 2K resolution (FID 51.73 vs. 54.60) while providing a $3.8\times$ speedup (20s vs. 76s). Qualitative results in Fig. 9(A) further confirm that PixelRush seamlessly upscales 1K FLUX outputs to 2K with crisp high-frequency details, validating our pipeline as a model-agnostic plug-and-play module.

C.2. Naive Few-Step Baseline

A natural question is whether existing patch-based methods can simply be accelerated by replacing their backbone with a few-step model. To investigate this, we apply DemoFusion [7] with SDXL-Turbo under both full-noise and partial-noise inversion settings. As shown in Fig. 9(B), this naive integration fails catastrophically in both full noise and partial noise inversion settings (top and bottom examples), producing severe artifacts. This is because standard patch-based methods rely on a long denoising trajectory (50 steps) to gradually reconstruct structure and harmonize details across patches. Simply reducing the number of steps breaks this process. In contrast, PixelRush is specifically designed for the few-step regime and maintains coherence throughout.

Overlap Fraction	# patches	4096 × 4096		
		FID (↓)	IS (↑)	Time (sec)
0.5	49	54.67	13.75	20
0.25	25	54.28	13.71	16

Table 7. **Analysis of the patch overlap fraction on 4K image generation.** A smaller overlap of 25% significantly reduces the number of patches, leading to faster inference

Figure 10. **Qualitative comparison of different patch overlap fractions.** This comparison shows that reducing the patch overlap from 50% to 25% yields a speedup and produces a visually indistinguishable result.

C.3. Comparison with Super-Resolution

As demonstrated in prior work [7, 23], traditional super-resolution methods have been shown to underperform compared to training-free diffusion pipelines like FreeScale [23] and DemoFusion [7]. Consequently, we exclude these methods from our baselines to focus on the state-of-the-art training-free paradigms.

C.4. Reducing Overlap Fraction To Speed Up

Following prior work [7], we partition the latent space into patches with an overlap of 50%. We note that the necessity for such a large overlap in previous methods often stems from their use of stochastic q-sampling, which requires a wide blending region to average out inconsistencies. In contrast, our use of deterministic DDIM-inv preserves the global structure better. This is a potential for optimization, as a smaller overlap could potentially increase inference speed without a significant loss in quality.

The number of patches can be quantified. The number of patches n required along one spatial dimension is related to the upscale factor M, and the overlap fraction o by the formula:

$$n - (n-1) \cdot o = M \tag{5}$$

For example, to generate a 4096×4096 image ($M = 4$), a 50% overlap ($o = 0.5$) requires $n = 7$, for a total of $n^2 = 49$ patches. However, the formula shows that when using a 25% overlap, we just need $n = 5$, reducing the number of patches to only 25 patches. As shown empirically in Tab. 7 and Fig. 10, reducing the patch count is an optimization for our pipeline, yielding a faster inference process with no loss in perceptual quality.

λ	# steps	2048 × 2048		
		FID (↓)	IS (↑)	Time (sec)
0.7	1	58.57	13.72	4
0.8	1	52.57	14.09	4
0.9	1	50.02	14.15	4
0.95	1	50.13	14.32	4

Table 8. **Analysis of the noise injection coefficient λ.** We analyze the sensitivity of our method to the choice of λ for $2K$ image generation. We identify $\lambda = 0.95$ as the optimal setting, yielding the best FID and IS scores.

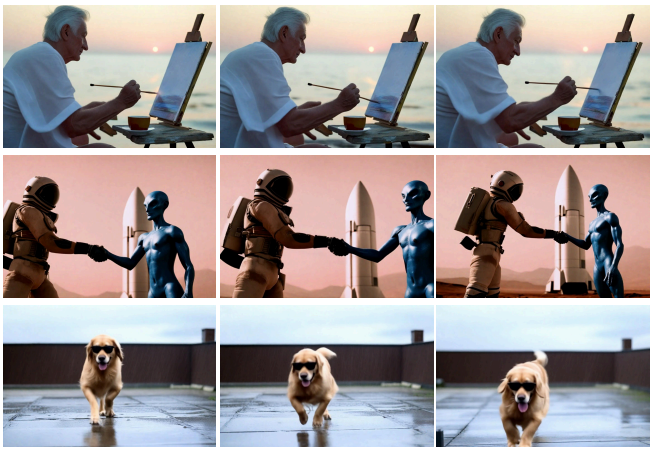

Figure 11. **Extending PixelRush to high-resolution video synthesis.** The figure showcases consecutive frames from videos upscaled frame-by-frame with our method. PixelRush successfully enhances the detail and sharpness in each frame, demonstrating its applicability to video content. Best viewed **ZOOMED-IN**.

C.5. Ablation On Noise Injection

To analyze the sensitivity of our method to the noise injection coefficient λ, we performed an ablation across a range of values. The results presented in Tab. 8 identify $\lambda = 0.95$ as the optimal setting, which we use throughout our work.

D. Additional Qualitative Results

D.1. Extension To High-Resolution Video Generation

Our method can be directly extended to high-resolution video generation by applying it frame-by-frame. We apply PixelRush independently to each frame of low-resolution

video generated by CogVideoX-5B. As demonstrated in Fig. 11, this approach successfully enhances the resolution and synthesize fine-grained details within each individual frame. However, because our method is applied on a per-frame fashion, it does not enforce temporal consistency across the video sequence, could result in flickering artifacts between frames. Improving temporal coherence presents a potential direction for future research.

D.2. Flexible Aspect Ratio Image Generation

Our PixelRush pipeline is capable of generating images across a wide range of aspect ratios, as shown in Figure 12.

D.3. A Gallery of Extreme-Resolution Synthesis

To demonstrate the superiority of PixelRush, we test its performance at extreme resolutions by generating $8K$ and $16K$ images. As shown in the Fig. 13, Fig. 14 and Fig. 15, our method synthesizes images with crisp, fine-grained details and exceptional perceptual quality. This is achieved with remarkable efficiency compared to other methods. For $8K$ image generation, PixelRush requires approximately 100 seconds, whereas competing methods can take up to 5400 seconds (90 minutes) for the same task. Furthermore, PixelRush can generate a $16K$ image in just 395 seconds, a resolution that is computationally infeasible for most other training-free approaches.

Figure 12. **PixelRush supports the generation of images with diverse aspect ratios.** Best viewed **ZOOMED-IN**.

Figure 13. **16K Qualitative Results.** Best viewed **ZOOMED-IN**.

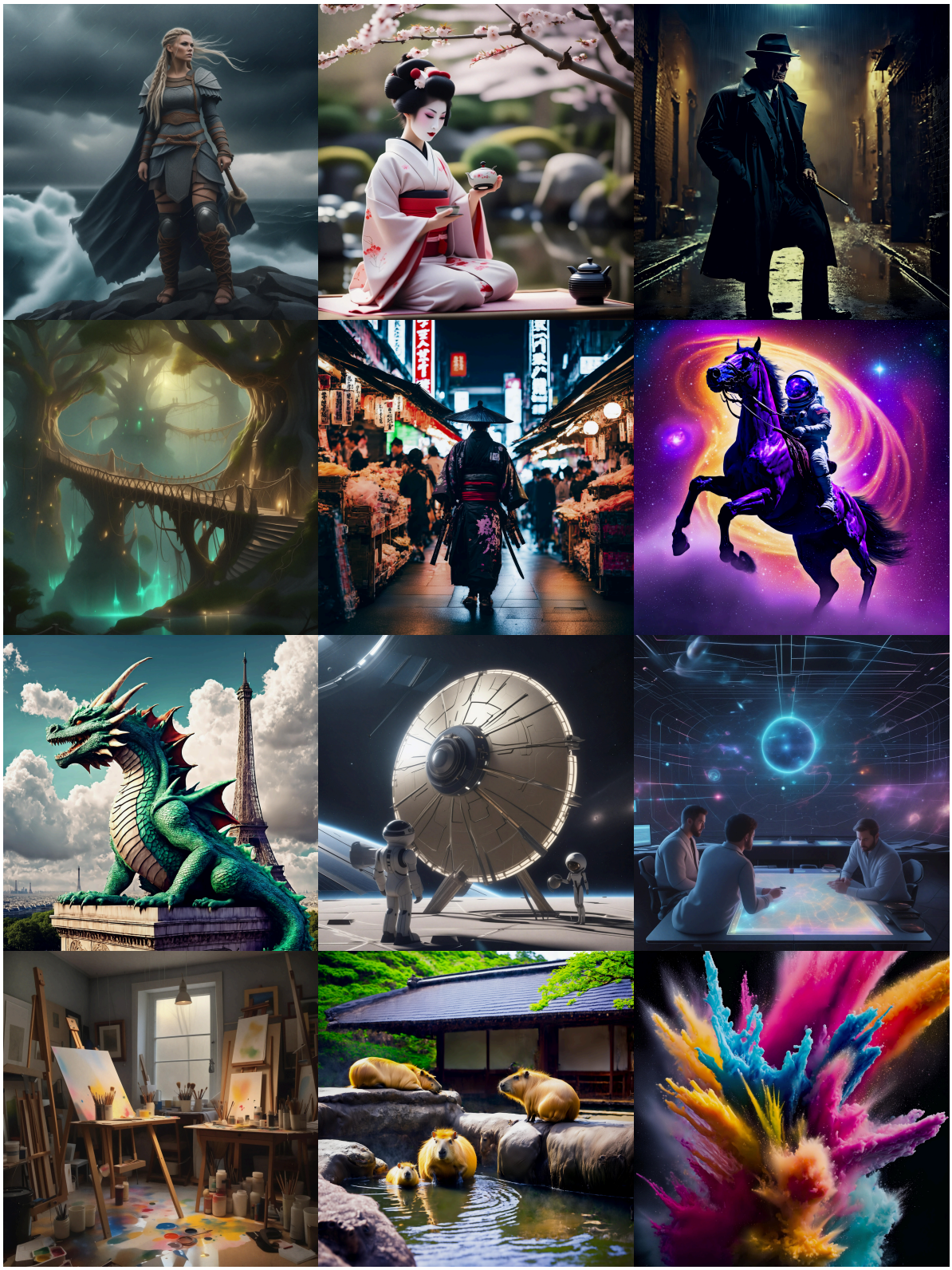

Figure 14. **8K Qualitative Results.** Best viewed **ZOOMED-IN**.

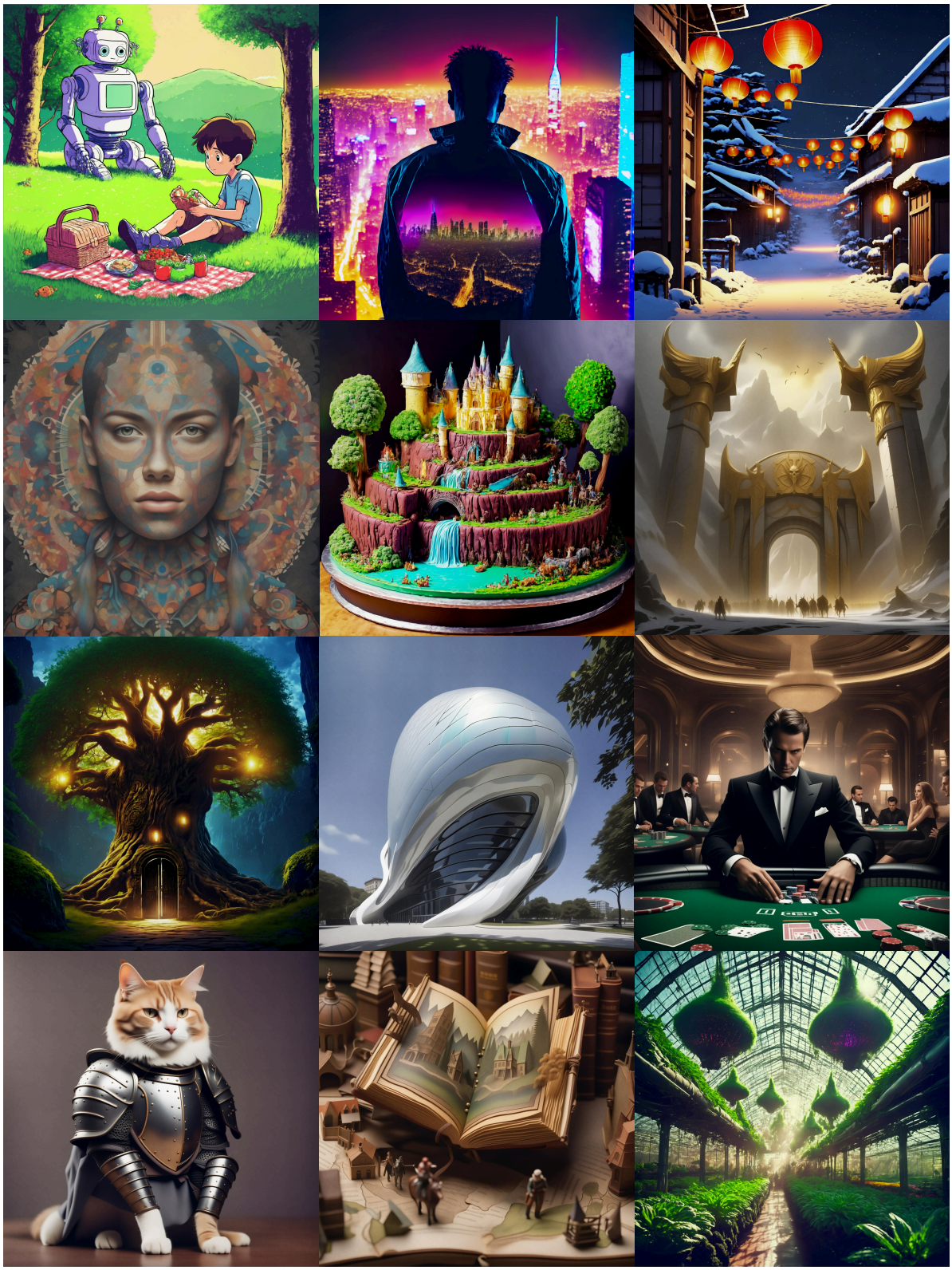

Figure 15. **8K Qualitative Results.** Best viewed **ZOOMED-IN**.