JANO: Adaptive Diffusion Generation with Early-stage Convergence Awareness

Supplementary Material

1. DDIM and Flow Matching

In DDIM, the denoiser is trained in the epsilon–prediction regime under the forward path

$$x_t = \alpha(t)\, x_{\text{data}} + \sigma(t)\, \varepsilon, \qquad \alpha(t) = \sqrt{\bar{\alpha}_t},\ \ \sigma(t) = \sqrt{1-\bar{\alpha}_t},$$

and the network output satisfies the MMSE identity

$$\hat{\varepsilon}_\theta(x_t, t) = \mathbb{E}[\varepsilon \,|\, x_t].$$

Flow matching targets the conditional mean instantaneous velocity along the same path,

$$v^*(x_t, t) = \mathbb{E}[\dot{x}_t \,|\, x_t], \qquad \dot{x}_t = \dot{\alpha}(t)\, x_{\text{data}} + \dot{\sigma}(t)\, \varepsilon.$$

Using

$$x_t = \alpha(t)\, \mathbb{E}[x_{\text{data}} \,|\, x_t] + \sigma(t)\, \mathbb{E}[\varepsilon \,|\, x_t],$$

where $\mathbb{E}[x_{\text{data}} \,|\, x_t] = \frac{x_t - \sigma(t)\, \hat{\varepsilon}_\theta(x_t,t)}{\alpha(t)}$. the DDIM epsilon output can be deterministically converted to the flow–matching velocity field:

$$\hat{v}_{\text{FM}}(x_t, t) = \frac{\dot{\alpha}(t)}{\alpha(t)}\, x_t + \left(\dot{\sigma}(t) - \frac{\dot{\alpha}(t)\, \sigma(t)}{\alpha(t)} \right) \hat{\varepsilon}_\theta(x_t, t). \quad (1)$$

In discrete schedulers, the derivatives are obtained by finite differences between adjacent indices,

$$\dot{\alpha}(t) \approx \frac{\alpha(t) - \alpha(t-\Delta t)}{\Delta t}, \qquad \dot{\sigma}(t) \approx \frac{\sigma(t) - \sigma(t-\Delta t)}{\Delta t},$$

After applying this transformation, the resulting estimates demonstrate properties consistent with flow matching theory. We validate this empirically using the DDIM sampler on Latte [8], as illustrated in Fig. 1.

2. Experimental Parameters

For the results reported in the main text, JANO is configured with the parameter settings listed in Table 1, and the warmup steps are the same as those used in Section 6.3. To separately assess image and video generation quality, we conduct experiments on prompts from the t2i-diversity-evalprompts dataset [1] for image generation and from the VBench benchmark [4] for video generation.

3. Additional Related Work

Several recent works have explored region-aware approaches in diffusion models, though their primary focus differs from JANO's early-stage acceleration objective.

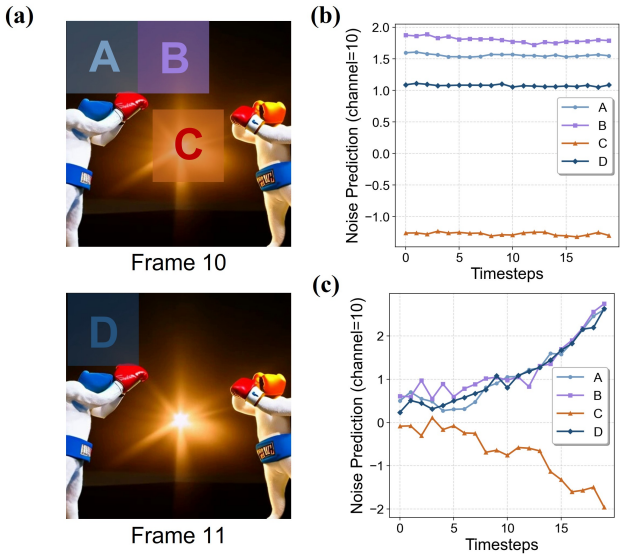

Figure 1. Latte video generation. (a) Frames 10 and 11 from the Latte-generated clip; markers A–D indicate four probe locations—the same A–D indices used in panels (b) and (c). (b) Raw DDIM model output at A–D across reverse-diffusion timesteps, before any transformation. (c) The same signals after applying the epsilon-to-FM linear mapping (Eq. (1)).

Upsample What Matters [5] proposes a mixed-resolution approach for DiT refinement by performing additional sampling on selected regions after low-resolution generation completion. Region-Aware Diffusion Models [6] and AdaDiffSR [3] incorporate region awareness for specific applications—image inpainting and super-resolution respectively. While these works demonstrate the value of region-based processing, they operate on existing generated content where region identification is more straightforward compared to JANO's challenge of early-stage noisy region assessment.

It's worth noting that these approaches fundamentally differ from JANO as they perform region-aware processing after initial generation or on partially complete content. In contrast, RAS [7] shares our goal of accelerating inference through dynamic region identification, which we analyze in detail in the following section.

4. Comparison with Concurrent Work: RAS

In this section, we provide a comparative analysis with the concurrent work, Region-Aware Sampler (RAS) [7], to clarify the fundamental conceptual differences and superior

Table 1. JANO parameter settings and performance.

Model	Static		Moderate		Speedup ↑	LPIPS ↓	SSIM ↑	PSNR ↑
	threshold	interval	threshold	interval				
Flux-1	0.1	8	0.5	5	1.86	0.1310	0.8533	24.40
	0.2	10	0.5	3	2.00	0.1728	0.8182	23.11
	0.3	10	0.6	3	2.07	0.1768	0.8153	23.08
Wan-1.3B	0.1	3	0.3	2	1.45	0.0588	0.8799	24.31
	0.15	6	0.4	4	2.09	0.1001	0.8325	21.71
	0.4	6	0.6	4	2.25	0.1033	0.8305	21.71
Wan-14B	0.1	6	0.3	3	1.71	0.0640	0.8723	30.73
	0.15	6	0.4	4	2.02	0.0732	0.8646	29.27
	0.4	6	0.6	4	2.30	0.0737	0.8658	28.87

performance of JANO. Both our work and RAS address the same fundamental challenge: accelerating diffusion models by adaptively allocating computation across different regions. However, despite this shared high-level objective, our approaches are built on fundamentally different philosophies and technical foundations.

RAS adopts a **reactive, observation-driven strategy**. It is predicated on the intuition that a model's computational focus shifts throughout the generation process. Consequently, it relies on continuous, step-by-step monitoring, employing a simple heuristic—the variance of the predicted noise in a patch—to decide which regions to update. This methodology is limited in two critical ways: first, the underlying observation of a "shifting focus" is not rigorously validated. Second, the chosen heuristic is too simplistic to accurately capture a region's true computational demand. This results in imprecise region identification and significantly inferior performance.

In stark contrast, JANO introduces a **proactive, theory-driven paradigm**. Our work is grounded in a more fundamental principle, which we extensively motivate and verify in Section 3: a region's inherent content complexity is directly correlated with its convergence requirements. JANO leverages this insight to foresee the entire computational trajectory for every region. This prediction is made possible by a theoretically robust metric derived from flow-matching principles (detailed in Section 4.2) and is accomplished in a single shot within the early stages of generation.

Therefore, the core novelty of JANO lies in its demonstration, both theoretically and empirically, that regional computational requirements can be **predicted** during early generation stages. This early-stage awareness enables JANO to precisely optimize resource allocation, resulting in superior acceleration while maintaining high generation quality across both image and video generation tasks on much newer diffusion models.

4.1. Empirical Comparison

To quantitatively validate the advantages of our approach, we implemented JANO and compared it with RAS's official implementation on Stable Diffusion 3 [2] under identical settings. As demonstrated in Fig. 2, JANO consistently outperforms RAS in both generation speed and output quality, corroborating our theoretical analysis.

Method	Times (s) ↓	LPIPS ↓	SSIM ↑	PSNR ↑
JANO	2.12 (1.63×)	0.066	0.940	26.03
RAS	2.53 (1.34×)	0.226	0.768	15.43

Figure 2. Performance comparison between JANO and RAS on Stable Diffusion 3, showing generated images and latency-quality analysis. Test prompt: "A photorealistic cute cat wearing a simple blue shirt, standing against a clear sky background."

Region Selection Strategy RAS is essentially designed to identify, at each diffusion timestep, the locally emphasized regions of generation, whereas JANO aims to infer the global image complexity already at early stages of the process. Fig. 3 illustrates this fundamental distinction: for RAS, regions with low prediction variance correspond to the tokens the model is currently focusing on, but at timestep 5 it still fails to capture the overall image complexity, and the identified regions drift as the timestep evolves. In contrast, JANO is able to reliably estimate the global complexity of the target image as early as timestep 5.

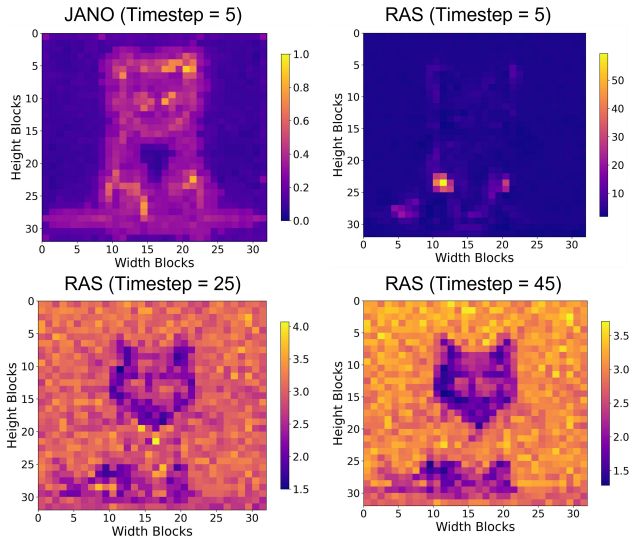

Figure 3. Comparison of JANO complexity maps and RAS token-importance maps at different diffusion timesteps.

5. Memory Overhead

JANO requires additional memory to store KV outputs for static and moderate tokens at each layer, introducing notable memory overhead. Table 2 shows the memory consumption before and after enabling KV cache:

Table 2. Memory consumption comparison with different caching strategies.

Model	Base Memory	With KV Cache
Flux-1	32.47 GB	34.94 GB
Wan-1.3B	7.57 GB	28.18 GB
Wan-14B	65.02 GB	269.02 GB

As model size and sequence length increase, the KV cache memory footprint becomes significant. To address this, we implement two optimization strategies: (1) reducing memory usage to 1/4 through conditional-free guidance parallelism and hidden state caching instead of KV caching for Wan-14B; (2) developing an asynchronous stream-based KV cache CPU offloading mechanism that enables single-GPU execution of Wan-14B with only 10% additional latency compared to the non-offloaded version.

6. Visualization of JANO

We provide visualization comparison between original video generation and JANO on Flux-1 and Wan 2.1, as shown in Fig. 4, Fig 5 and Fig. 6. We conduct both Text-to-Image generation and Text-to-Video generation under model-default resolution. Results demonstrates JANO can preserve high pixel-level fidelity, achieving similar generation quality compared with the original generation.

References

[1] Manuel Brack, Sudeep Katakol, Felix Friedrich, Patrick Schramowski, Hareesh Ravi, Kristian Kersting, and Ajinkya Kale. How to train your text-to-image model: Evaluating design choices for synthetic training captions. *arXiv preprint arXiv:2506.16679*, 2025. 1

[2] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis, 2024. 2

[3] Yuanting Fan, Chengxu Liu, Nengzhong Yin, Changlong Gao, and Xueming Qian. Adadiffsr: Adaptive region-aware dynamic acceleration diffusion model for real-world image super-resolution, 2024. 1

[4] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 1

[5] Wongi Jeong, Kyungryeol Lee, Hoigi Seo, and Se Young Chun. Upsample what matters: Region-adaptive latent sampling for accelerated diffusion transformers, 2025. 1

[6] Sora Kim, Sungho Suh, and Minsik Lee. Rad: Region-aware diffusion models for image inpainting, 2024. 1

[7] Ziming Liu, Yifan Yang, Chengruidong Zhang, Yiqi Zhang, Lili Qiu, Yang You, and Yuqing Yang. Region-adaptive sampling for diffusion transformers, 2025. 1

[8] Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *CoRR*, abs/2401.03048, 2024. 1

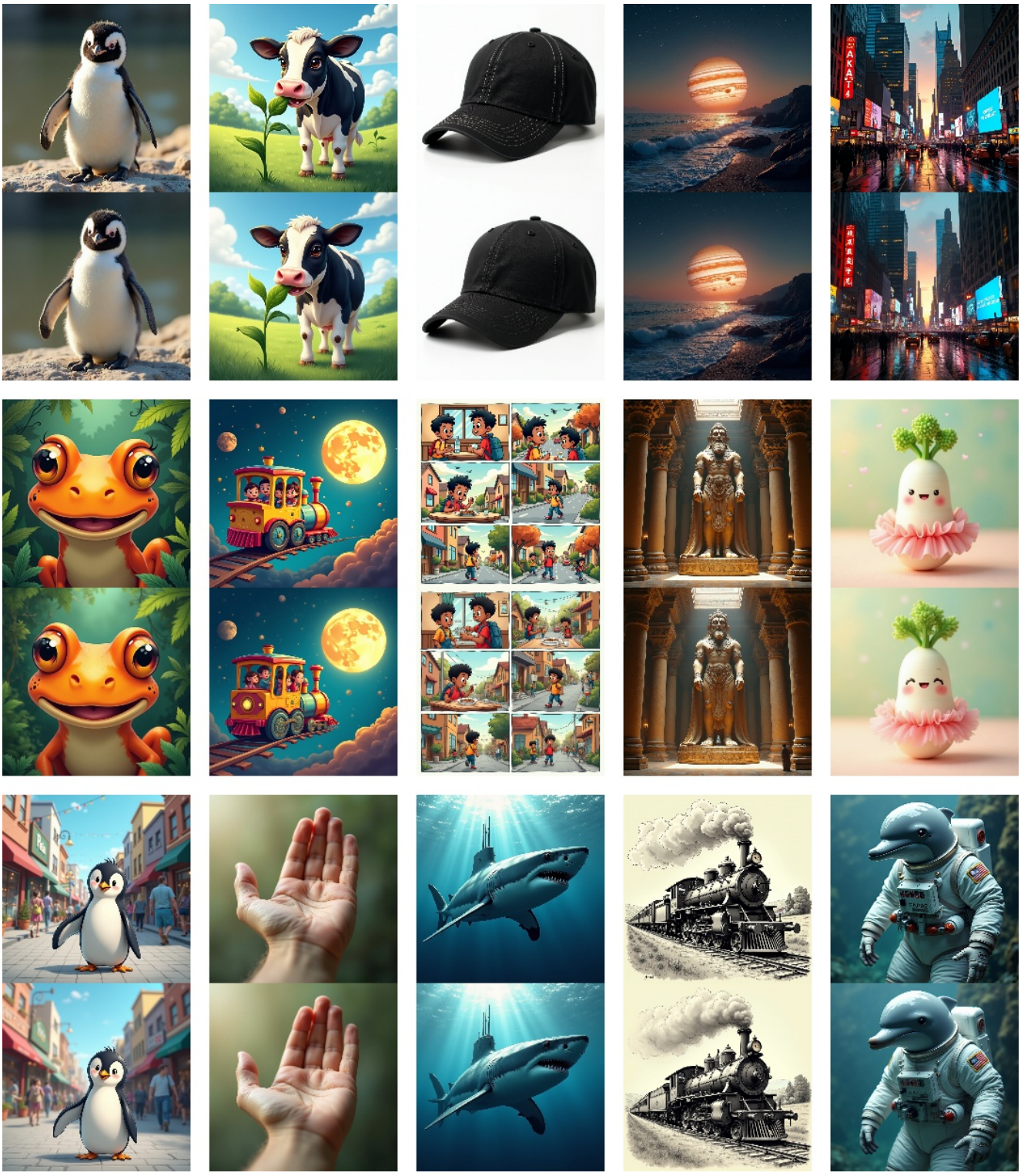

Figure 4. Flux visualization comparison between original images (top) and JANO results (bottom).

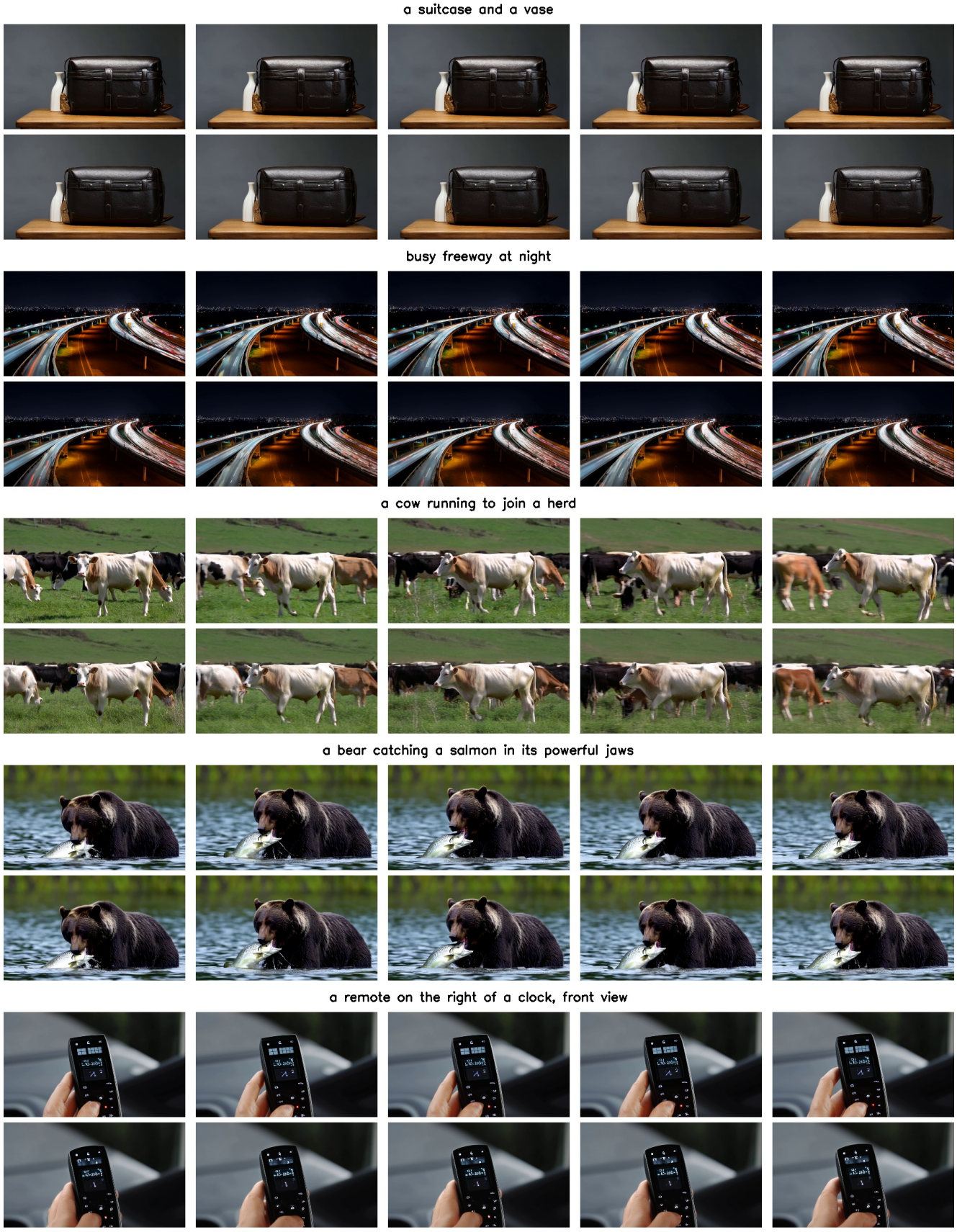

Figure 5. Wan-1.3B visualization comparison between original videos (top) and JANO results (bottom).

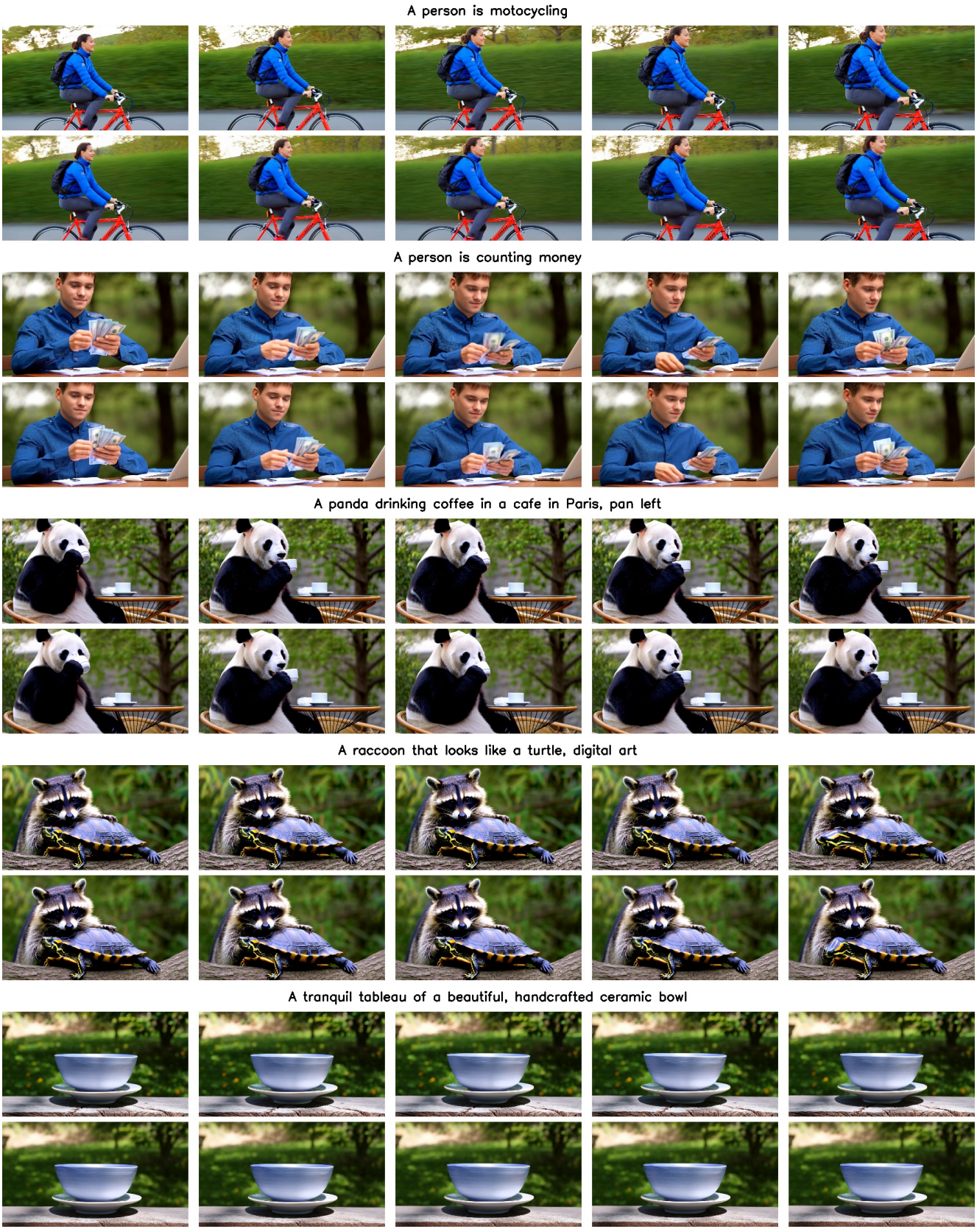

Figure 6. Wan-14B visualization comparison between original videos (top) and JANO results (bottom).