Efficient Training-Free High-Resolution Synthesis with Energy Rectification in Diffusion Models

Zhen Yang[1*]   Guibao Shen[1*]   Minyang Li[1]   Liang Hou[2]   Mushui Liu[4]
Luozhou Wang[1]   Xin Tao[2]   Pengfei Wan[2]   Di Zhang[2]   Ying-Cong Chen[1,3†]
[1]HKUST(GZ)   [2]Kuaishou Technology   [3]HKUST   [4]Zhejiang University
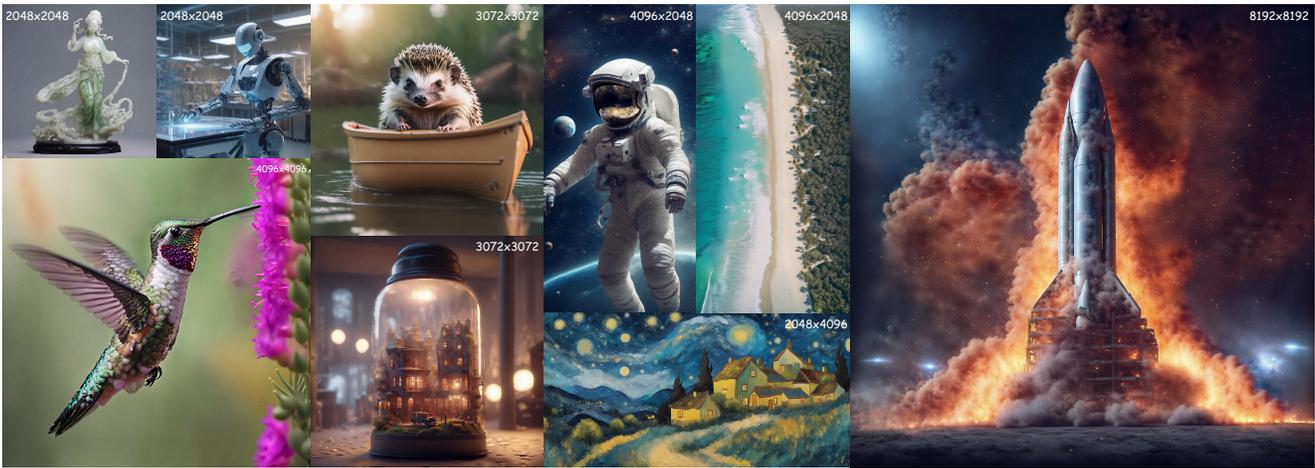zheny.cs@gmail.com, yingcongchen@ust.hk

Figure 1. Generated images by *RectifiedHR*. The training-free *RectifiedHR* enables diffusion models (SDXL is shown in the figure) to synthesize images at resolutions exceeding their original training resolution. Please zoom in for a closer view.

Abstract

Diffusion models have achieved remarkable progress across various visual generation tasks. However, their performance significantly declines when generating content at resolutions higher than those used during training. Although numerous methods have been proposed to enable high-resolution generation, they all suffer from inefficiency. In this paper, we propose RectifiedHR, a straightforward and efficient solution for training-free high-resolution synthesis. Specifically, we propose a noise refresh strategy that unlocks the model's training-free high-resolution synthesis capability and improves efficiency. Additionally, we are the first to observe the phenomenon of energy decay, which may cause image blurriness during the high-resolution synthesis process. To address this issue, we introduce average latent energy analysis and find that tuning the classifier-free guidance hyperparameter can significantly improve generation performance. Our method is entirely training-free and demonstrates efficient performance. Furthermore, we show that RectifiedHR is compatible with various diffusion model techniques, enabling advanced features such as image editing, customized generation, and video synthesis. Extensive comparisons with numerous baseline methods validate the superior effectiveness and efficiency of RectifiedHR. The project page can be found here.

1. Introduction

Recent advances in diffusion models [6, 11, 28, 31, 35, 38, 42, 45, 62] have significantly improved generation quality, enabling realistic editing [1, 4, 8, 26, 40, 41, 55, 58] and customized generation [2, 9, 12, 30, 47, 54]. However, these models struggle to generate images at resolutions beyond those seen during training, resulting in noticeable performance degradation. Training directly on high-resolution content is computationally expensive, underscoring the need for methods that enhance resolution without requiring additional training.

1

Figure 2. The visualization images corresponding to "predicted x_0" at different time step t, abbreviated as $p_{x_0}^t$. The figure visualizes the process of how $p_{x_0}^t$ changes with the sampling steps, where the x-axis represents the timestep in the sampling process. The 11 images are evenly extracted from 50 steps. It can be observed that in the first half of the process, $p_{x_0}^t$ is mainly responsible for global structure generation, while the second half is mainly responsible for local detail generation. Moreover, later in the sampling process, the image corresponding to $p_{x_0}^t$ exhibits the characteristics of an RGB image.

Currently, the most naive approach is to directly input high-resolution noise. However, this method leads to severe repeated pattern issues. To address this problem, many training-free high-resolution generation methods have been proposed, such as [2, 5, 10, 13–15, 21, 22, 24, 27, 29, 32, 33, 37, 50, 57, 60, 61]. However, these methods all share a common problem: they inevitably introduce additional computational overhead. For example, the sliding window operations introduced by [2, 10, 22, 29, 32, 33] have overlapping regions that result in redundant computations. Similarly, [33, 37, 50] require setting different prompts for small local regions of each image and need to incorporate a vision-language model. Additionally, [5, 27, 61] require multiple rounds of SDEdit [39] or complex classifier-free guidance (CFG) to gradually increase the resolution from a low-resolution image to a high-resolution image, thereby introducing more sampling steps or complex CFG calculations. All of these methods introduce additional computational overhead and complexity, significantly reducing the speed of high-resolution synthesis.

We propose an efficient framework, *RectifiedHR*, to enable high-resolution synthesis by progressively increasing resolution during sampling. The simplest baseline is to progressively increase the resolution in the latent space. However, naive resizing in latent space introduces noise and artifacts. We identify two critical issues and propose corresponding solutions: (1) Since the latent space is obtained by transforming RGB images via a VAE, RGB-based resizing becomes invalid in the latent space (Tab. 2, Method D). Moreover, as the latent comprises "predicted x_0" and Gaussian noise, direct resizing distorts the noise distribution. To address this, we propose noise refresh, which independently resizes "predicted x_0"—shown to exhibit RGB characteristics in late sampling (Fig. 2)—and injects fresh noise to maintain a valid latent distribution while increasing resolution. (2) We are the first to observe that resizing "predicted x_0": introduces spatial correlations, reducing pixel-wise independence, causing detail loss and blur, and leading to energy decay (Fig. 3a). To mitigate this, we propose energy rectification, which adjusts the CFG hyperparameter (Fig. 3b) to compensate for the energy decay and effectively eliminate blur. Compared to [5, 27, 61], our method
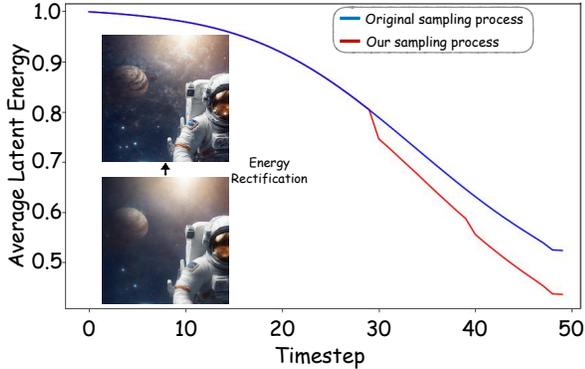
achieves high-resolution synthesis without additional sampling steps or complex CFG calculations, ensuring computational efficiency.

In general, our main contributions are as follows: (1) We propose *RectifiedHR*, an efficient, training-free framework for high-resolution synthesis that eliminates redundant computation and enables resolution scalability without requiring additional sampling steps. (2) We introduce noise refresh and energy rectification, pioneering the use of average latent energy analysis to address energy decay—an issue previously overlooked in high-resolution synthesis. (3) Our method surpasses existing baselines in both efficiency and quality, achieving faster inference while preserving superior fidelity. (4) We demonstrate that *RectifiedHR* can be seamlessly integrated with ControlNet, supporting a range of applications such as image editing, customized image generation, and video synthesis.
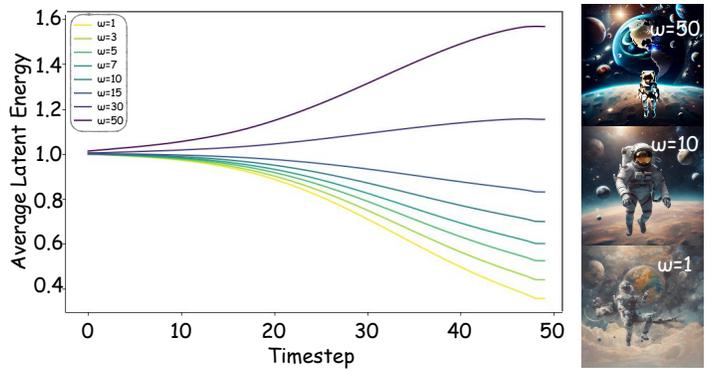
2. Related Work

2.1. Text-guided image generation

With the scaling of models, data volume, and computational resources, text-guided image generation has witnessed unprecedented advancements, leading to the emergence of numerous diffusion models such as LDM [45], SDXL [42], PixArt [6, 7], HunyuanDiT [31], LuminaNext [62], FLUX [28], SD3 [11], LCM [38], and UltraPixel [44]. These models learn mappings from Gaussian noise to high-quality images through diverse training and sampling strategies, including DDPM [19], SGM [52], EDM [25], DDIM [51], flow matching [34], rectified flow [36], RDM [53], and pyramidal flow [23]. However, these methods typically require retraining and access to high-resolution datasets to support high-resolution generation. Consequently, exploring training-free approaches for high-resolution synthesis has become a key area of interest within the vision generation community. Our method is primarily designed to enable efficient, training-free high-resolution synthesis in a plug-and-play manner.

2

(a) The energy decay phenomenon of our noise refresh sampling process is evaluated in comparison to the original sampling process across 100 random prompts.

(b) The evolution of average latent energy over timesteps during the generation of 1024×1024 resolution images from 100 random prompts under different classifier-free guidance hyperparameters.

Figure 3. (a) The x-axis denotes the timesteps of the sampling process, and the y-axis indicates the average latent energy. The blue line shows the average latent energy of the original sampling process when generating 1024×1024-resolution images. The red line corresponds to our noise refresh sampling process, where noise refresh is applied at the 30th and 40th timesteps, and the resolution progressively increases from 1024×1024 to 2048×2048, and subsequently to 3072×3072. It can be observed that noise refresh induces a noticeable decay in average latent energy. From the left images, it is evident that after energy rectification, image details become more pronounced. (b) The x-axis represents the timestep, the y-axis represents the average latent energy, and ω denotes the hyperparameter for classifier-free guidance. It can be observed that the average latent energy increases as ω increases. From the right figures, one can observe how the generated images vary with increasing ω.

2.2. Training-free high-resolution image generation

Due to the domain gap across different resolutions, directly applying diffusion models to high-resolution image generation often results in pattern repetition and poor semantic structure. MultiDiffusion [2] proposes a sliding window denoising scheme for panoramic image generation. However, this method suffers from severe pattern repetition, as it primarily focuses on the aggregation of local information. Improved variants based on the sliding window denoising scheme include SyncDiffusion [29], Demofusion [10], AccDiffusion [33], and CutDiffusion [32]. Specifically, SyncDiffusion incorporates global information by leveraging the gradient of perceptual loss from the predicted denoised images at each denoising step as guidance. Demofusion employs progressive upscaling, skip residuals, and dilated sampling mechanisms to support higher-resolution image generation. AccDiffusion introduces patch-content-aware prompts, while CutDiffusion adopts a coarse-to-fine strategy to mitigate pattern repetition. Nonetheless, these approaches share complex implementation logic and encounter efficiency bottlenecks due to redundant computation arising from overlapping sliding windows.

ScaleCrafter [15], FouriScale [21], HiDiffusion [60], and Attn-SF [24] modify the network architecture of the diffusion model, which may result in suboptimal performance. Furthermore, these methods perform high-resolution denoising throughout the entire sampling process, leading to slower inference compared to our approach, which progressively transitions from low to high resolution. Although HiDiffusion accelerates inference using window attention mechanisms, our method remains faster, as demonstrated by experimental results.

Upscale Guidance [22] and ElasticDiffusion [14] both propose incorporating global and local denoising information into classifier-free guidance [18]. The global branch of Upscale Guidance and the overlapping window regions in the local branch of ElasticDiffusion involve significantly higher computational complexity compared to our progressive resolution increase strategy. ResMaster [50] and HiPrompt [37] introduce multi-modal models to regenerate prompts and enrich image details; however, the use of such multi-modal models introduces substantial overhead, leading to further efficiency issues.

DiffuseHigh [27], MegaFusion [57], FreCas [61], and AP-LDM [5] leverage the detail enhancement capabilities of SDEdit [39], progressively adding details from low-resolution to high-resolution images. In contrast to these methods, our approach neither increases sampling steps nor requires additional computations involving classifier-free guidance (CFG) variants, resulting in greater efficiency. Moreover, we identify the issue of energy decay and show that simply adjusting the classifier-free guidance parameter is sufficient to rectify the energy and achieve improved results.

3

Algorithm 1 Original Sampling Process

Require: $x_T \sim \mathcal{N}(0, I), 0 \leq \omega \in \mathbb{R}$
1: **for** i in range(50) **do**
2: $\quad \tilde{\epsilon}(x_t, t, \emptyset) = \hat{\epsilon}(x_t, t, c) + \omega \cdot [\hat{\epsilon}(x_t, t, c) - \hat{\epsilon}(x_t, t, \emptyset)]$
3: $\quad p^t_{x_0} \leftarrow (x_t - \sqrt{1 - \bar{\alpha}_t}\tilde{\epsilon}(x_t, t))/\sqrt{\bar{\alpha}_t}$
4: $\quad x_{t-1} = \sqrt{\bar{\alpha}_{t-1}}p^t_{x_0} + \sqrt{1 - \bar{\alpha}_{t-1}}\tilde{\epsilon}(x_t, t)$
5: **end for**
6: **return** x_0

Algorithm 2 Our Sampling Process

Require: $x_T \sim \mathcal{N}(0, I), 0 \leq \omega \in \mathbb{R}$
Require: $\Omega_{rectified} = \{\{T_i : \omega_i\}|i = 1...N - 1\}$
1: **for** i in range(50) **do**
2: $\quad \tilde{\epsilon}(x_t, t) = \hat{\epsilon}(x_t, t, \emptyset) + \omega \cdot [\hat{\epsilon}(x_t, t, c) - \hat{\epsilon}(x_t, t, \emptyset)]$
3: $\quad p^t_{x_0} \leftarrow (x_t - \sqrt{1 - \bar{\alpha}_t}\tilde{\epsilon}(x_t, t))/\sqrt{\bar{\alpha}_t}$
4: $\quad$ **if** i in $\Omega_{rectified}$.keys() **then**
5: $\quad\quad \tilde{p}^t_{x_0} \leftarrow E(resize(D(p^t_{x_0})))$
6: $\quad\quad \epsilon \sim \mathcal{N}(0, I_{\tilde{p}^t_{x_0}})$
7: $\quad\quad x_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\tilde{p}^t_{x_0} + \sqrt{1 - \bar{\alpha}_{t-1}}\epsilon$
8: $\quad\quad \omega = \Omega_{rectified}[t]$
9: $\quad$ **else**
10: $\quad\quad x_{t-1} = \sqrt{\bar{\alpha}_{t-1}}p^t_{x_0} + \sqrt{1 - \bar{\alpha}_{t-1}}\tilde{\epsilon}(x_t, t)$
11: $\quad$ **end if**
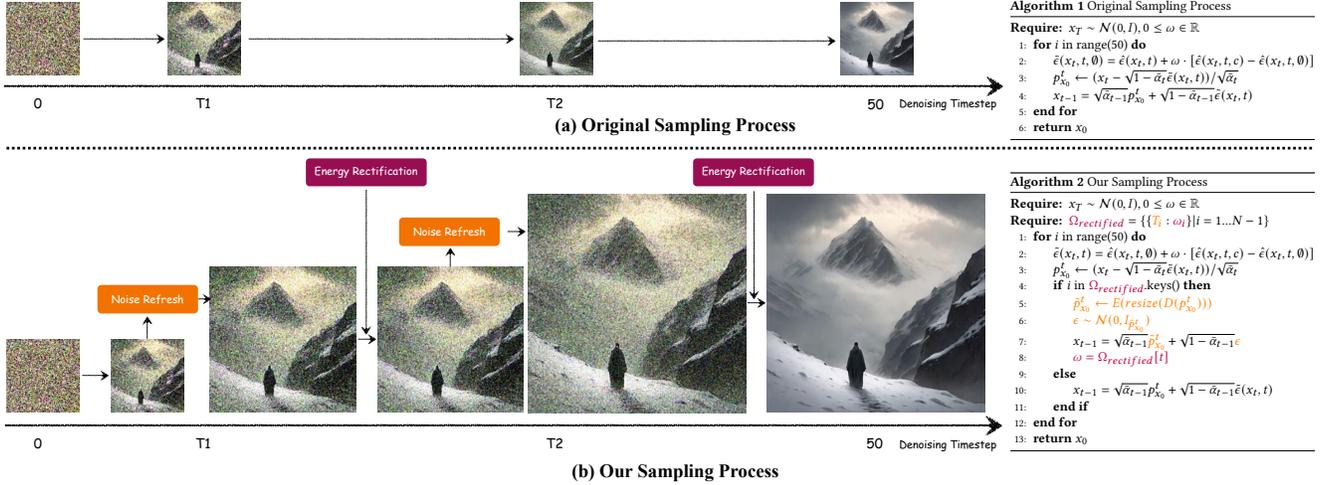12: **end for**
13: **return** x_0

Figure 4. Overview of *RectifiedHR*. (a) The original sampling process and its pseudocode. (b) The sampling process and pseudocode of our method. The orange components in the pseudocode and modules correspond to **Noise Refresh**, while the purple components represent **Energy Rectification**. ϵ denotes Gaussian random noise, whose shape adapts to that of $\tilde{p}^t_{x_0}$. The definitions of other symbols used in the pseudocode can be found in Sec. 3.1.

3. Method

3.1. Preliminaries

Diffusion models establish a mapping between Gaussian noise and images, enabling image generation by randomly sampling noise. In this paper, we assume 50 sampling steps, with the denoising process starting at step 0 and ending at step 49. We define I_o as the RGB image. During training, the diffusion model first employs a VAE encoder $E(\cdot)$ to transform the RGB image into a lower-dimensional latent representation, denoted as x_0. The forward diffusion process is then defined as:

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon. \quad (1)$$

Noise of varying intensity is added to x_0 to produce different x_t, where $\bar{\alpha}_t$ is a time-dependent scheduler parameter controlling the noise strength, and ϵ is randomly sampled Gaussian noise. The diffusion model $\hat{\epsilon}(x_t, t, c)$, parameterized by θ, is optimized to predict the added noise via the following training objective:

$$\min_{\theta} \mathbb{E}_{x_t, t, c}\left[\|\epsilon - \hat{\epsilon}(x_t, t, c)\|_2^2\right], \quad (2)$$

where c denotes the conditioning signal for generation (e.g., a text prompt in T2I tasks). During inference, random noise is sampled in the latent space, and the diffusion model gradually transforms this noise into an image via a denoising process. Finally, the latent representation is passed through the decoder $D(\cdot)$ of the VAE to reconstruct the generated RGB image. The objective of high-resolution synthesis is to produce images at resolutions beyond those seen during training—for instance, resolutions exceeding 1024×1024 in our setting.

Classifier-free guidance for diffusion models. Classifier-free guidance (CFG) [18] is currently widely adopted to enhance the quality of generated images by incorporating unconditional outputs at each denoising step. The formulation of classifier-free guidance is as follows:

$$\tilde{\epsilon}(x_t, t) = \hat{\epsilon}(x_t, t, \emptyset) + \omega \cdot [\hat{\epsilon}(x_t, t, c) - \hat{\epsilon}(x_t, t, \emptyset)], \quad (3)$$

where ω is the hyperparameter of classifier-free guidance, $\hat{\epsilon}(x_t, t, \emptyset)$ and $\hat{\epsilon}(x_t, t, c)$ denote the predicted noises from the unconditional and conditional branches, respectively. We refer to $\tilde{\epsilon}(x_t, t)$ as the predicted noise after applying classifier-free guidance.

Sampling process for diffusion models. In this paper, we adopt the DDIM sampler [51] as the default. The deterministic sampling formulation of DDIM is given as follows:

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\underbrace{\left(\frac{x_t - \sqrt{1 - \bar{\alpha}_t} \cdot \tilde{\epsilon}(x_t, t)}{\sqrt{\bar{\alpha}_t}}\right)}_{\text{predicted } x_0 \rightarrow p^t_{x_0}} \\ + \sqrt{1 - \bar{\alpha}_{t-1}} \cdot \tilde{\epsilon}(x_t, t). \quad (4)$$

As illustrated in Eq. 4, at timestep t, we first predict the noise $\tilde{\epsilon}(x_t, t)$ using the pre-trained neural network $\hat{\epsilon}(\cdot)$. We then compute a "predicted x_0" at timestep t, denoted as $p^t_{x_0}$. Finally, x_{t-1} is derived from $\tilde{\epsilon}(x_t, t)$ and $p^t_{x_0}$ using the diffusion process defined in Eq. 4.

In this paper, we propose RectifiedHR, which consists of noise refresh and energy rectification. The noise refresh module progressively increases the resolution during

4

the sampling process, while the energy rectification module enhances the visual details of the generated contents.

3.2. Noise refresh

To enable high-resolution synthesis, we propose a progressive resizing strategy during sampling. A straightforward baseline for implementing this strategy is to directly perform image-space interpolation in the latent space. However, this approach presents two key issues. First, since the latent space is obtained via VAE compression of the image, interpolation operations that work in RGB space are ineffective in the latent space, as demonstrated by Method D in the ablation study (Table 2). Second, because the latent space consists of $p_{x_0}^t$ and noise, directly resizing it alters the noise distribution, potentially shifting the latent representation outside the diffusion model's valid domain. To address this, we visualize $p_{x_0}^t$, as shown in Fig. 2, and observe that the image corresponding to $p_{x_0}^t$ exhibits RGB-like characteristics in the later stages of sampling. Therefore, we resize $p_{x_0}^t$ to enlarge the latent representation. To ensure the resized latent maintains a Gaussian distribution, we inject new Gaussian noise into $p_{x_0}^t$. The method for enhancing the resolution of $p_{x_0}^t$ is as follows:

$$\tilde{p}_{x_0}^t = E(\text{resize}(D(p_{x_0}^t))), \qquad (5)$$

where E denotes the VAE encoder, D denotes the VAE decoder, and $\text{resize}(\cdot)$ refers to the operation of enlarging the RGB image. We adopt bilinear interpolation as the default resizing method. The procedure for re-adding noise is as follows:

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\tilde{p}_{x_0}^t + \sqrt{1 - \bar{\alpha}_{t-1}}\epsilon, \qquad (6)$$

where ϵ denotes a random Gaussian noise that shares the same shape as $\tilde{p}_{x_0}^t$. We refer to this process as **Noise Refresh**.

As illustrated in Fig. 4b, the noise refresh operation is applied at specific time points T_i during the sampling process. To automate the selection of these timesteps T, we propose the following selection formula:

$$T_i = \lfloor (T_{\max} - T_{\min}) * (\frac{i-1}{N})^{M_T} + T_{\min} \rfloor, \qquad (7)$$

where $T_{\max}$ and $T_{\min}$ define the range of sampling timesteps at which noise refresh is applied. N denotes the number of different resolutions in the denoising process, and $N-1$ corresponds to the number of noise refresh operations performed. N is a positive integer, and the range of i includes all integers in $[1, N)$. Specifically, we set T_0 to 0 and $T_{\max}$ to the total number of sampling steps. $T_{\min}$ is treated as a hyperparameter. Since $p_{x_0}^t$ exhibits more prominent image features in the later stages of sampling, as shown in Fig. 2, $T_{\min}$ is selected to fall within the later stage of the sampling process. A quantitative analysis of the variation in $p_{x_0}^t$ is provided in Supplementary 7.1.

3.3. Energy rectification

Although noise refresh enables the diffusion model to generate high-resolution images, we observe that introducing noise refresh during the sampling process leads to blurriness in the generated content, as illustrated in the fourth row of Fig. 6. To investigate the cause of this phenomenon, we introduce the average latent energy formula as follows:

$$\mathbb{E}[x_t^2] = \frac{\sum_{i=1}^{C}\sum_{j=1}^{H}\sum_{k=1}^{W} x_{t_{ijk}}^2}{C \times H \times W}, \qquad (8)$$

where x_t represents the latent variable at time t, and C, H, and W denote the channel, height, and width dimensions of the latent, respectively. This definition closely resembles that of image energy and is used to quantify the average energy per element of the latent vector. To investigate the issue of image blurring, we conduct an average latent energy analysis on 100 random prompts. As illustrated in Fig. 3a, we first compare the average latent energy between the noise refresh sampling process and the original sampling process. We observe significant energy decay during the noise refresh sampling process, which explains why the naive implementation produces noticeably blurred images. Subsequently, we experimentally discover that the hyperparameter ω in classifier-free guidance influences the average latent energy. As shown in Fig. 3b, we find that increasing the classifier-free guidance parameter ω leads to a gradual increase in energy. Therefore, the issue of energy decay—and thus image quality degradation—can be mitigated by increasing ω to boost the energy in the noise refresh sampling scheme. As demonstrated in the left image of Fig. 3a, once energy is rectified by using a larger classifier-free guidance hyperparameter ω, the blurriness is substantially reduced, and the generated image exhibits significantly improved clarity. We refer to this process of correcting energy decay as **Energy Rectification**. However, we note that a larger ω is not always beneficial, as excessively high values may lead to overexposure. The goal of energy rectification is to align the energy level with that of the original diffusion model's denoising process, rather than to maximize energy indiscriminately. The experiment analyzing the rectified average latent energy curve is provided in Supplementary 7.6.

As shown in Fig. 4b, the energy rectification operation is applied during the sampling process following noise refresh. To automatically select an appropriate ω value for classifier-free guidance, we propose the following selection formula:

$$\omega_i = (\omega_{\max} - \omega_{\min}) * (\frac{i}{N-1})^{M_\omega} + \omega_{\min}, \qquad (9)$$

where $\omega_{\max}$ and $\omega_{\min}$ define the range of ω values used in classifier-free guidance during the sampling process. N denotes the number of different resolutions in the denoising

Figure 5. Qualitative comparison between our method and SDXL+BSRGAN at a resolution of 2048×2048.

	Methods	FID$_r$↓	KID$_r$↓	IS$_r$↑	FID$_c$↓	KID$_c$↓	IS$_c$↑	CLIP↑	Time↓
2048 × 2048	FouriScale	71.344	0.010	15.957	53.990	0.014	20.625	31.157	59s
	ScaleCrafter	64.236	0.007	15.952	45.861	0.010	22.252	31.803	35s
	HiDiffusion	63.674	0.007	16.876	41.930	0.008	23.165	31.711	18s
	CutDiffusion	59.152	0.007	17.109	38.004	0.008	23.444	32.573	53s
	ElasticDiffusion	56.639	0.010	15.326	37.649	0.014	19.867	32.301	150s
	AccDiffusion	48.143	**0.002**	18.466	32.747	0.008	24.778	33.153	111s
	DiffuseHigh	49.748	0.003	19.537	27.667	0.004	27.876	33.436	37s
	FreCas	49.129	0.003	20.274	27.002	0.004	**29.843**	33.700	14s
	DemoFusion	**47.079**	**0.002**	19.533	26.441	0.004	27.843	33.748	79s
	Ours	48.361	**0.002**	20.616	25.347	**0.003**	28.126	**33.756**	**13s**
4096 × 4096	FouriScale	135.111	0.046	9.481	129.895	0.057	9.792	26.891	489s
	ScaleCrafter	110.094	0.028	10.098	112.105	0.043	11.421	27.809	528s
	HiDiffusion	93.515	0.024	11.878	120.170	0.058	11.272	27.853	71s
	CutDiffusion	130.207	0.055	9.334	113.033	0.055	10.961	26.734	193s
	ElasticDiffusion	101.313	0.056	9.406	111.102	0.089	7.627	27.725	400s
	AccDiffusion	54.918	0.005	17.444	60.362	0.023	16.370	32.438	826s
	DiffuseHigh	48.861	**0.003**	19.716	40.267	0.010	21.550	33.390	190s
	FreCas	49.764	**0.003**	18.656	39.047	0.010	**21.700**	33.237	74s
	DemoFusion	48.983	**0.003**	18.225	38.136	0.010	20.786	33.311	605s
	Ours	**48.684**	**0.003**	20.352	**35.718**	**0.009**	20.819	**33.415**	**37s**

Table 1. Comparison to SOTA methods at 2048×2048 and 4096×4096 resolutions. Bold numbers indicate the best performance, while underlined numbers denote the second-best performance. ↑ and ↓ represent metrics where higher and lower values are better, respectively. The subscript r refers to resizing high-resolution images to 299×299 before evaluation, whereas the subscript c indicates that 10 patches of size 1024×1024 are randomly cropped from each generated high-resolution image and then resized to 299×299 for evaluation. The detailed definitions of the metrics are provided in Supplementary 7.7.

process, and $N-1$ corresponds to the number of noise refresh operations performed. N is a positive integer, and the range of i includes all integers in $[0, N)$. $\omega_{\min}$ refers to the CFG hyperparameter at the original resolution supported by the diffusion model. M_ω is a tunable hyperparameter that allows for different strategies in selecting ω_i. The value of N used in Eq. 7 and Eq. 9 remains consistent throughout the sampling process.

Additionally, we establish the connection between energy rectification and SNR correction strategies proposed in [20, 57, 61], showing that SNR correction is essentially a form of energy rectification. The complete proof is provided in Supplementary 7.2.

4. Experiments

4.1. Evaluation Setup

Our experiments use SDXL [42] as the base model, which by default generates images at a resolution of 1024×1024. Furthermore, our method can also be applied to Stable Diffusion and transformer-based diffusion models such as WAN [56] and SD3 [11], as demonstrated in Fig. 8 and Supplementary 7.3. The specific evaluation metrics and methods are provided in Supplementary 7.7. The comparison includes state-of-the-art training-free methods: Demofusion [10], DiffuseHigh [27], HiDiffusion [60], CutDiffusion [32], ElasticDiffusion [14], FreCas [61], FouriScale [21], ScaleCrafter [15], and AccDiffusion [33]. Quantitative assessments focus on upsampling to 2048×2048 and 4096×4096 resolutions. All baseline methods are fairly and fully reproduced. For the 2048×2048 resolution setting, we set $T_{\min}$ to 40, $T_{\max}$ to 50, N to 2, $\omega_{\min}$ to 5, $\omega_{\max}$ to 30, M_T to 1, and M_ω to 1. For the 4096×4096 resolution setting, we set $T_{\min}$ to 40, $T_{\max}$ to 50, N to 3, $\omega_{\min}$ to 5, $\omega_{\max}$ to 50, M_T to 0.5, and M_ω to 0.5. All experiments are conducted using 8 NVIDIA A800 GPUs unless specified. The above hyperparameters are obtained through a hyperparameter search, with detailed ablation studies provided in Supplementary 7.4.

4.2. Quantitative Results

As shown in Tab. 1, our proposed method, *RectifiedHR*, consistently outperforms competing approaches in both the 2048×2048 and 4096×4096 resolution settings. Specifically, in the 2048×2048 setting, *RectifiedHR* achieves the highest scores in 6 out of 8 evaluated metrics, ranks second in one, and third in another. In the 4096×4096 setting, *RectifiedHR* attains the highest scores in 7 out of 8 metrics and ranks third in the remaining one. In the 2048×2048 setting, our KID_r ranks third, primarily due to the fact that this metric requires resizing high-resolution images to a lower resolution for evaluation, which inadequately captures fine-grained high-resolution details. This limitation has been noted in previous works such as [10, 33]. Although our method ranks second and third in IS_c, the superior performance across the remaining metrics, along with high computational efficiency, demonstrates the overall effectiveness of *RectifiedHR*. These results underscore the robustness and efficacy of our method for high-resolution image generation tasks.

Furthermore, when scaled to a resolution of 4096×4096, *RectifiedHR* exhibits exceptional computational efficiency, operating at approximately twice the speed of the next fastest competitor. This significant speed advantage stems from our strategy of preserving the original number of sampling steps while optimizing performance through careful
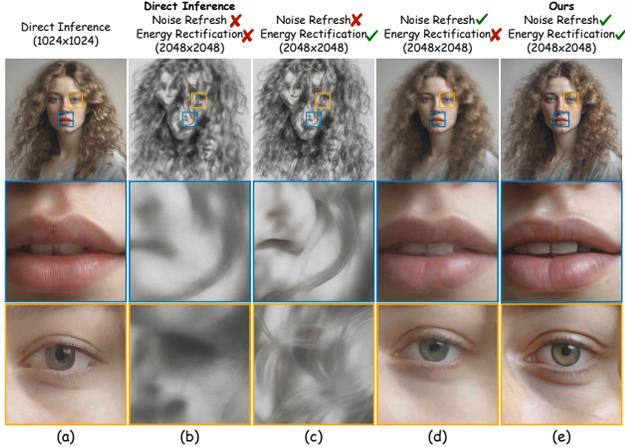
Figure 6. Qualitative results of the ablation studies at 2048×2048 resolution. The orange and blue boxes indicate enlarged views of local regions within the high-resolution image. Zoom in for details.

	Methods	Noise Refresh	Energy Rectification	Resize Latent	$FID_r \downarrow$	$KID_r \downarrow$	$IS_r \uparrow$	$FID_c \downarrow$	$KID_c \downarrow$	$IS_c \uparrow$	CLIP $\uparrow$
2048×2048	A	×	×	×	98.676	0.030	13.193	73.426	0.029	17.867	30.021
	B	×	✓	×	86.595	0.021	13.900	60.625	0.021	19.921	30.728
	C	✓	×	×	79.743	0.021	13.334	76.023	0.035	11.840	29.966
	D	×	✓	✓	78.307	0.019	13.221	74.419	0.034	11.883	29.523
	Ours	✓	✓	×	**48.361**	**0.002**	**20.616**	**25.347**	**0.003**	**28.126**	**33.756**
4096×4096	A	×	×	×	187.667	0.088	8.636	111.117	0.057	13.383	25.447
	B	×	✓	×	175.830	0.079	8.403	80.733	0.034	15.791	26.099
	C	✓	×	×	85.088	0.026	13.114	141.422	0.091	5.465	29.548
	D	×	✓	✓	89.968	0.033	11.973	145.472	0.103	6.312	28.212
	Ours	✓	✓	×	**48.684**	**0.003**	**20.352**	**35.718**	**0.009**	**20.819**	**33.415**

Table 2. Quantitative results of the ablation studies. Method A denotes direct inference (without noise refresh and energy rectification), Method B excludes noise refresh, Method C excludes energy rectification, and Method D replaces noise refresh in our method with direct latent resizing. Ours refers to the full version of our proposed method. The detailed definitions of the evaluation metrics are provided in Supplementary 7.7.

tuning of the CFG hyperparameter. In contrast, alternative methods such as DiffuseHigh introduce substantial computational overhead by incorporating additional sampling steps via repeated applications of techniques like SDEdit and FreCas within more computationally intensive CFG frameworks. Notably, *RectifiedHR* achieves this superior speed without compromising image quality, consistently producing high-resolution outputs with visual fidelity that meets or exceeds that of baseline methods across evaluated resolutions. These results highlight *RectifiedHR*'s effective balance between speed and quality, reinforcing its efficiency and practicality for high-resolution synthesis.

4.3. Qualitative Results

As shown in Fig. 7, to clearly illustrate the differences between our method and existing baselines, we select a representative prompt for each of the three resolution scenarios

and conduct qualitative comparisons against SDXL direct inference, AccDiffusion, DemoFusion, FouriScale, FreCas, HiDiffusion, and ScaleCrafter. AccDiffusion and DemoFusion tend to produce blurry details and lower visual quality, such as the peacock's eyes and feathers in column b, and the bottle stoppers in column c. FouriScale and ScaleCrafter often generate deformed or blurred objects that fail to satisfy the prompt, such as feathers lacking peacock characteristics in column b, and a blurry bottle body missing the velvet element specified in the prompt in column c. HiDiffusion may introduce repetitive patterns, as seen in the duplicate heads in column b and the recurring motifs on the bottles in column c. FreCas can produce distorted details or fail to adhere to the prompt, such as the deformed and incorrect number of bottles in column c. In contrast, our method consistently achieves superior visual quality across all resolutions. In column a, our approach generates the clearest and most refined faces and is the only method that correctly captures the prompt's description of the sun and moon intertwined. In column b, our peacock is the most detailed and visually accurate, with a color distribution and fine-grained features that closely align with the prompt's reference to crystal eyes and delicate feather-like gears. In column c, our method demonstrates the highest fidelity in rendering the bottle stopper and floral patterns, and it uniquely preserves the white velvet background described in the prompt. These qualitative results highlight the effectiveness of our method in generating visually consistent, detailed, and prompt-faithful images across different resolution settings.

4.4. Comparison with the super-resolution model

Training-free high-resolution image generation methods primarily exploit intrinsic properties of diffusion models to achieve super-resolution. Beyond the aforementioned approaches, another viable strategy adopts a two-stage pipeline that combines diffusion models with dedicated super-resolution models. For example, methods such as SDXL + BSRGAN first generate an image using a diffusion model, then apply a super-resolution model to upscale it to the target resolution. To further evaluate the differences between SDXL+BSRGAN and our method, we conduct additional qualitative comparisons. The experimental setup follows that described in Sec. 4.1. As shown in Fig. 5, we observe that when images generated by SDXL exceed the domain of the original training data—such as in cases involving distorted facial features—BSRGAN is unable to correct these artifacts, resulting in performance degradation. Furthermore, existing two-stage approaches rely on pre-trained super-resolution models constrained by fixed-resolution training data. In contrast, our method inherently adapts to arbitrary resolutions without retraining. For example, as demonstrated in the 2048×4096 resolution scene in
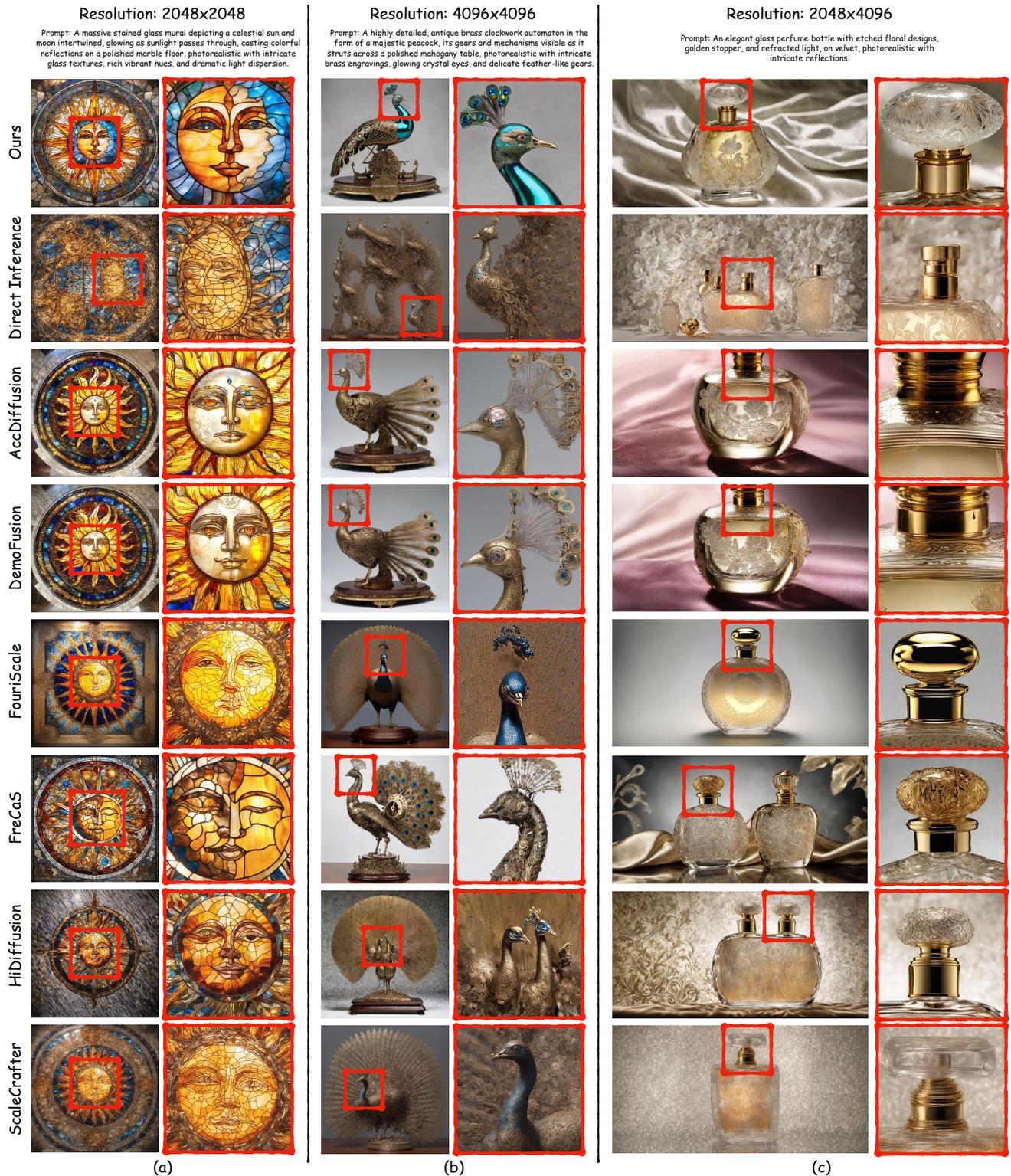
Figure 7. Qualitative comparison across three different resolutions between our method and other training-free methods. The red box indicates an enlarged view of a local region within the high-resolution image.
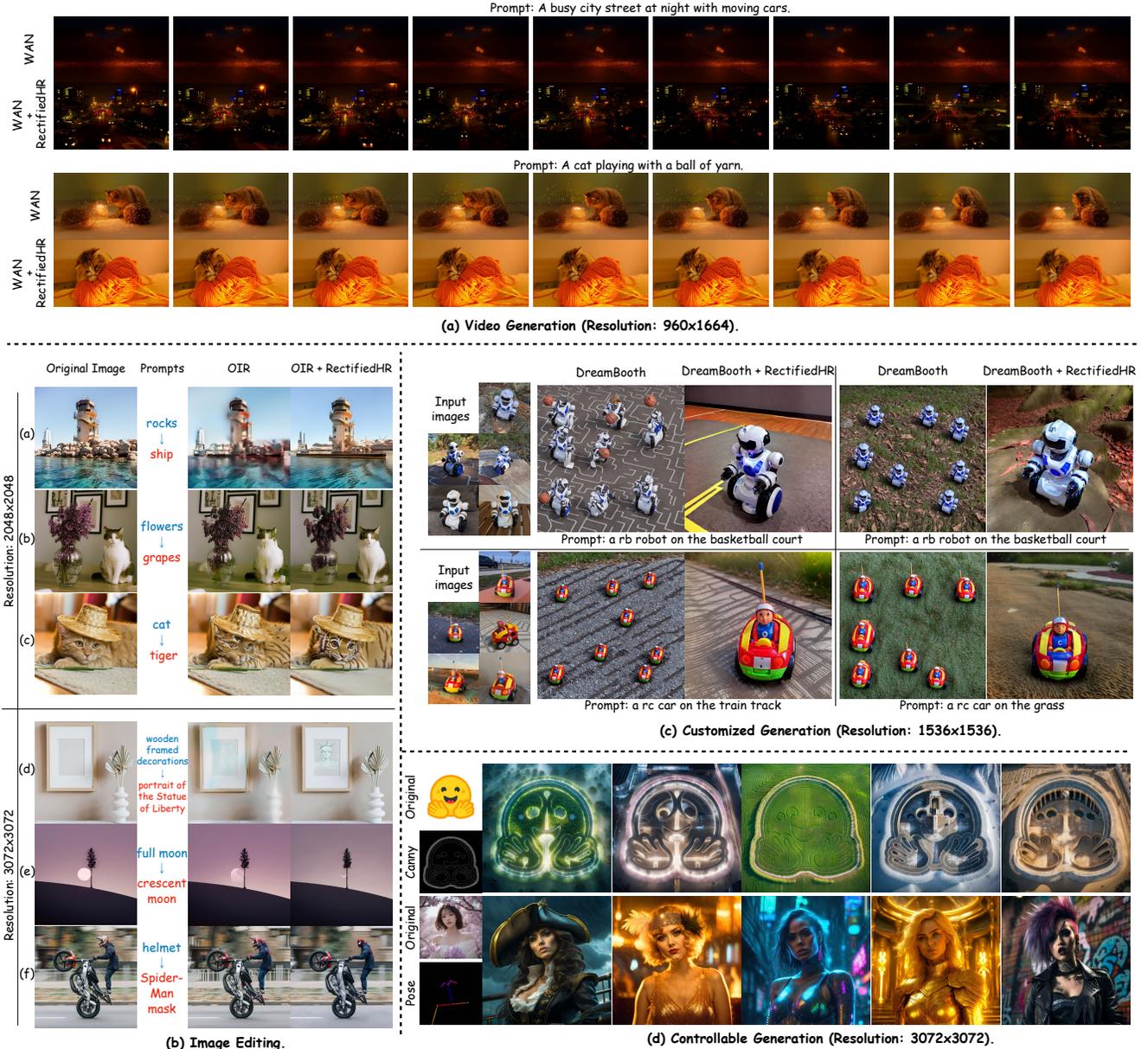
(a) Video Generation (Resolution: 960×1664).

(b) Image Editing.

(c) Customized Generation (Resolution: 1536×1536).

(d) Controllable Generation (Resolution: 3072×3072).

Figure 8. Applications. (a) Results of integrating *RectifiedHR* with the video diffusion model WAN 1.3B (which supports a default resolution of 480×832). Videos are generated at a resolution of 960×1664. (b) Results of integrating *RectifiedHR* with the image editing method OIR using SDXL (which supports a default resolution of 1024×1024). Images are edited at resolutions of 2048×2048 or 3072×3072. (c) Results of integrating *RectifiedHR* with DreamBooth using Stable Diffusion v1.4 (which supports a default resolution of 512×512). Images are generated at a resolution of 1536×1536. (d) Results of integrating *RectifiedHR* with ControlNet using SDXL (which supports a default resolution of 1024×1024). Images are generated at a resolution of 3072×3072. **Contents are best viewed when zoomed in.**

Fig. 7, our approach remains effective, whereas BSRGAN cannot be applied.

4.5. Ablation Study

To evaluate the effectiveness of each module in our method, we conduct both quantitative experiments (Tab. 2) and qual-

itative experiments (Fig. 6). The metric computation follows the procedure described in Supplementary 7.7. All hyperparameters are set according to Sec. 4.1. Additionally, in scenarios without energy rectification, the classifier-free guidance hyperparameter ω is fixed at 5. For simplicity, this section mainly compares the FID_c metric at the

9

4096×4096 resolution.

Comparing Method B in Tab.2 with Ours, the FID_c increases from 35.718 to 80.733 without noise refresh. As shown in Fig. 6c vs. Fig. 6e, this performance drop is due to the failure in generating correct semantic structures caused by the absence of noise refresh. Fig. 6d and Fig. 6e highlight the critical role of energy rectification in enhancing fine details. Comparing Method C in Tab. 2 with Ours, the FID_c rises sharply from 35.718 to 141.422 without energy rectification, demonstrating that energy decay severely degrades generation quality. This underscores the importance of energy rectification—despite its simplicity, it yields significant improvements. Comparing Method D in Tab. 2 with Ours, the FID_c improves from 145.472 to 35.718, revealing that directly resizing the latent is ineffective. This confirms that noise refresh is indispensable and cannot be replaced by naïve latent resizing.

We also conduct ablation studies on the hyperparameters related to Eq. 7 and Eq. 9, with detailed results provided in Supplementary 7.4.

5. More Applications

This section highlights how *RectifiedHR* can enhance a variety of tasks, with a focus on demonstrating visual improvements. The experiments cover diverse tasks, models, and sampling methods to validate the effectiveness of our approach. While primarily evaluated on classic methods and models, *RectifiedHR* can also be seamlessly integrated into more advanced techniques. Supplementary 7.5 provides detailed quantitative results and corresponding hyperparameter settings for reference.

Video Generation. *RectifiedHR* can be directly applied to video diffusion models such as WAN [56]. The officially supported maximum resolution for WAN 1.3B is 480×832. As shown in Fig. 8a, directly generating high-resolution videos with WAN may lead to generation failure or prompt misalignment. However, integrating *RectifiedHR* enables WAN to produce high-quality, high-resolution videos reliably.

Image Editing. *RectifiedHR* can also be applied to image editing tasks. In this section, we use SDXL as the base model with a default resolution of 1024×1024. Directly editing high-resolution images with OIR often results in ghosting artifacts, as illustrated in rows a, b, d, and e of Fig. 8b. Additionally, it can cause shape distortions and deformations, as shown in rows c and f. In contrast, the combination of OIR and *RectifiedHR* effectively mitigates these issues, as demonstrated in Fig. 8b.

Customized Generation. *RectifiedHR* can be directly adapted to DreamBooth using SD1.4 with a default resolution of 512×512, as shown in Fig. 8c. Direct generation of high-resolution customized images often leads to severe repetitive pattern artifacts. Integrating *RectifiedHR* effec-

tively addresses this problem.

Controllable Generation. *RectifiedHR* can be seamlessly integrated with ControlNet [59] using SDXL at a default resolution of 1024×1024 to enable controllable generation. As shown in Fig. 8d, control signals may include pose, canny edges, and other modalities.

6. Conclusion and Future Work

We propose an efficient and straightforward method, *RectifiedHR*, for high-resolution synthesis. Specifically, we conduct an average latent energy analysis and, to the best of our knowledge, are the first to identify the energy decay phenomenon during high-resolution synthesis. Our approach introduces a novel training-free pipeline that is both simple and effective, primarily incorporating noise refresh and energy rectification operations. Extensive comparisons demonstrate that *RectifiedHR* outperforms existing methods in both effectiveness and efficiency. Nonetheless, our method has certain limitations. During the noise refresh stage, it requires both decoding and encoding operations via the VAE, which impacts the overall runtime. In future work, we aim to investigate performing resizing operations directly in the latent space to further improve efficiency.

References

[1] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2live: Text-driven layered image and video editing. In *ECCV*, pages 707–723. Springer, 2022. 1

[2] Omer Bar-Tal, Lior Yariv, Yaron Lipman, and Tali Dekel. Multidiffusion: Fusing diffusion paths for controlled image generation. 2023. 1, 2, 3

[3] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401*, 2018. 19

[4] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, pages 18392–18402, 2023. 1

[5] Boyuan Cao, Jiaxin Ye, Yujie Wei, and Hongming Shan. Ap-ldm: Attentive and progressive latent diffusion model for training-free high-resolution image generation. *arXiv preprint arXiv:2410.06055*, 2024. 2, 3

[6] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-alpha: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023. 1, 2

[7] Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-sigma: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. In *European Conference on Computer Vision*, pages 74–91. Springer, 2025. 2

[8] Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. Diffedit: Diffusion-based semantic image editing with mask guidance. 2022. 1

[9] Ganggui Ding, Canyu Zhao, Wen Wang, Zhen Yang, Zide Liu, Hao Chen, and Chunhua Shen. Freecustom: Tuning-free customized image generation for multi-concept composition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9089–9098, 2024. 1

[10] 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. 2, 3, 6, 19

[11] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024. 1, 2, 6

[12] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022. 1

[13] 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. In *European Conference on Computer Vision*, pages 39–55. Springer, 2024. 2

[14] Moayed Haji-Ali, Guha Balakrishnan, and Vicente Ordonez. Elasticdiffusion: Training-free arbitrary size image generation through global-local content separation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6603–6612, 2024. 3, 6

[15] Yingqing He, Shaoshu Yang, Haoxin Chen, Xiaodong Cun, Menghan Xia, Yong Zhang, Xintao Wang, Ran He, Qifeng Chen, and Ying Shan. Scalecrafter: Tuning-free higher-resolution visual generation with diffusion models. In *The Twelfth International Conference on Learning Representations*, 2023. 2, 3, 6

[16] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. 2021. 14

[17] 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. 19

[18] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 3, 4

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2

[20] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *International Conference on Machine Learning*, pages 13213–13232. PMLR, 2023. 6, 15

[21] Linjiang Huang, Rongyao Fang, Aiping Zhang, Guanglu Song, Si Liu, Yu Liu, and Hongsheng Li. Fouriscale: A frequency perspective on training-free high-resolution image synthesis. In *European Conference on Computer Vision*, pages 196–212. Springer, 2025. 2, 3, 6

[22] Juno Hwang, Yong-Hyun Park, and Junghyo Jo. Upsample guidance: Scale up diffusion models without training. *arXiv preprint arXiv:2404.01709*, 2024. 2, 3, 15

[23] Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. *arXiv preprint arXiv:2410.05954*, 2024. 2

[24] Zhiyu Jin, Xuli Shen, Bin Li, and Xiangyang Xue. Training-free diffusion model adaptation for variable-sized text-to-image synthesis. *Advances in Neural Information Processing Systems*, 36:70847–70860, 2023. 2, 3

[25] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022. 2

[26] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *CVPR*, pages 6007–6017, 2023. 1

[27] Younghyun Kim, Geunmin Hwang, Junyu Zhang, and Eunbyung Park. Diffusehigh: Training-free progressive high-resolution image synthesis through structure guidance. *arXiv preprint arXiv:2406.18459*, 2024. 2, 3, 6

[28] Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2023. 1, 2

[29] Yuseung Lee, Kunho Kim, Hyunjin Kim, and Minhyuk Sung. Syncdiffusion: Coherent montage via synchronized joint diffusions. *Advances in Neural Information Processing Systems*, 36:50648–50660, 2023. 2, 3

[30] Dongxu Li, Junnan Li, and Steven Hoi. Blip-diffusion: Pre-trained subject representation for controllable text-to-image generation and editing. *Advances in Neural Information Processing Systems*, 36, 2024. 1

[31] Zhimin Li, Jianwei Zhang, Qin Lin, Jiangfeng Xiong, Yanxin Long, Xinchi Deng, Yingfang Zhang, Xingchao Liu, Minbin Huang, Zedong Xiao, et al. Hunyuan-dit: A powerful multi-resolution diffusion transformer with fine-grained chinese understanding. *arXiv preprint arXiv:2405.08748*, 2024. 1, 2

[32] Mingbao Lin, Zhihang Lin, Wengyi Zhan, Liujuan Cao, and Rongrong Ji. Cutdiffusion: A simple, fast, cheap, and strong diffusion extrapolation method. *arXiv preprint arXiv:2404.15141*, 2024. 2, 3, 6

[33] Zhihang Lin, Mingbao Lin, Meng Zhao, and Rongrong Ji. Accdiffusion: An accurate method for higher-resolution image generation. In *European Conference on Computer Vision*, pages 38–53. Springer, 2025. 2, 3, 6, 19

[34] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 2

[35] Mushui Liu, Yuhang Ma, Yang Zhen, Jun Dan, Yunlong Yu, Zeng Zhao, Zhipeng Hu, Bai Liu, and Changjie Fan. Llm4gen: Leveraging semantic representation of llms for text-to-image generation. *arXiv preprint arXiv:2407.00737*, 2024. 1

[36] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022. 2

[37] Xinyu Liu, Yingqing He, Lanqing Guo, Xiang Li, Bu Jin, Peng Li, Yan Li, Chi-Min Chan, Qifeng Chen, Wei Xue, et al. Hiprompt: Tuning-free higher-resolution generation with hierarchical mllm prompts. *arXiv preprint arXiv:2409.02919*, 2024. 2, 3

[38] Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. *arXiv preprint arXiv:2310.04378*, 2023. 1, 2

[39] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021. 2, 3

[40] Daiki Miyake, Akihiro Iohara, Yu Saito, and Toshiyuki Tanaka. Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models. 2023. 1

[41] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *CVPR*, pages 6038–6047, 2023. 1

[42] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 1, 2, 6

[43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 19

[44] Jingjing Ren, Wenbo Li, Haoyu Chen, Renjing Pei, Bin Shao, Yong Guo, Long Peng, Fenglong Song, and Lei Zhu. Ultrapixel: Advancing ultra-high-resolution image synthesis to new peaks. *arXiv preprint arXiv:2407.02158*, 2024. 2

[45] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1, 2

[46] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023. 18

[47] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Wei Wei, Tingbo Hou, Yael Pritch, Neal Wadhwa, Michael Rubinstein, and Kfir Aberman. Hyperdreambooth: Hypernetworks for fast personalization of text-to-image models. *arXiv preprint arXiv:2307.06949*, 2023. 1

[48] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016. 19

[49] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 14, 19

[50] Shuwei Shi, Wenbo Li, Yuechen Zhang, Jingwen He, Biao Gong, and Yinqiang Zheng. Resmaster: Mastering high-resolution image generation via structural and fine-grained guidance. *arXiv preprint arXiv:2406.16476*, 2024. 2, 3

[51] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2, 4

[52] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 2

[53] Jiayan Teng, Wendi Zheng, Ming Ding, Wenyi Hong, Jianqiao Wangni, Zhuoyi Yang, and Jie Tang. Relay diffusion: Unifying diffusion process across resolutions for image synthesis. *arXiv preprint arXiv:2309.03350*, 2023. 2

[54] Yoad Tewel, Rinon Gal, Gal Chechik, and Yuval Atzmon. Key-locked rank one editing for text-to-image personalization. In *ACM SIGGRAPH 2023 Conference Proceedings*, 2023. 1

[55] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *CVPR*, pages 1921–1930, 2023. 1

[56] Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 6, 10, 16

[57] Haoning Wu, Shaocheng Shen, Qiang Hu, Xiaoyun Zhang, Ya Zhang, and Yanfeng Wang. Megafusion: Extend diffusion models towards higher-resolution image generation without further tuning. *arXiv preprint arXiv:2408.11001*, 2024. 2, 3, 6, 15

[58] Zhen Yang, Ganggui Ding, Wen Wang, Hao Chen, Bohan Zhuang, and Chunhua Shen. Object-aware inversion and reassembly for image editing. *arXiv preprint arXiv:2310.12149*, 2023. 1, 17

[59] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023. 10, 18

[60] Shen Zhang, Zhaowei Chen, Zhenyu Zhao, Zhenyuan Chen, Yao Tang, Yuhao Chen, Wengang Cao, and Jiajun Liang. Hidiffusion: Unlocking high-resolution creativity and effi-

ciency in low-resolution trained diffusion models. *arXiv preprint arXiv:2311.17528*, 2023. 2, 3, 6

[61] Zhengqiang Zhang, Ruihuang Li, and Lei Zhang. Frecas: Efficient higher-resolution image generation via frequency-aware cascaded sampling. *arXiv preprint arXiv:2410.18410*, 2024. 2, 3, 6, 15

[62] Le Zhuo, Ruoyi Du, Han Xiao, Yangguang Li, Dongyang Liu, Rongjie Huang, Wenze Liu, Lirui Zhao, Fu-Yun Wang, Zhanyu Ma, et al. Lumina-next: Making lumina-t2x stronger and faster with next-dit. *arXiv preprint arXiv:2406.18583*, 2024. 1, 2

7. Supplementary

7.1. Quantitative Analysis of "Predicted x_0"

To quantitatively validate this observation, as shown in Fig.9, we conduct additional experiments on the generation of $p_{x_0}^t$ using 100 random prompts sampled from LAION-5B [49], and analyze the CLIP Score [16] and Mean Squared Error (MSE). From Fig. 9a, we observe that after 30 denoising steps, the MSE between $p_{x_0}^t$ and $p_{x_0}^{t-1}$ exhibits minimal change. In Fig. 9b, we find that the CLIP score between $p_{x_0}^t$ and the corresponding prompt increases slowly beyond 30 denoising steps.

Figure 9. The trend of the "predicted x_0" at different timesteps t, denoted as $p_{x_0}^t$, evaluated on 100 random prompts. (a) The average MSE between $p_{x_0}^t$ and $p_{x_0}^{t-1}$. The x-axis represents the sampling timestep, and the y-axis denotes the average MSE. It can be observed that after approximately 30 steps, the rate of change in $p_{x_0}^t$ slows significantly. (b) The trend of the average CLIP Score between $p_{x_0}^t$ and the prompt across different timesteps. The x-axis represents the sampling timestep, and the y-axis denotes the average CLIP Score.

7.2. The connection between energy rectification and Signal-to-Noise Ratio (SNR) correction

In the proof presented in this section, all symbols follow the definitions provided in the Method section of the main text. Any additional symbols not previously defined will be explicitly specified. This proof analyzes energy variation using the DDIM sampler as an example. The sampling formulation of DDIM is given as follows:

$$
\begin{aligned}
x_{t-1} &= \sqrt{\bar{\alpha}_{t-1}} \left(\frac{x_t - \sqrt{1 - \bar{\alpha}_t} \tilde{\epsilon}(x_t, t)}{\sqrt{\bar{\alpha}_t}} \right) + \sqrt{1 - \bar{\alpha}_{t-1}} \cdot \tilde{\epsilon}(x_t, t) \\
&= \sqrt{\frac{\bar{\alpha}_{t-1}}{\bar{\alpha}_t}} x_t + \left(\sqrt{1 - \bar{\alpha}_{t-1}} - \frac{\sqrt{\bar{\alpha}_{t-1}}\sqrt{1 - \bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}} \right) \tilde{\varepsilon}(x_t, t).
\end{aligned}
\tag{10}
$$

To simplify the derivation, we assume that all quantities in the equation are scalar values. Based on the definition of average latent energy in Eq.8 of the main text, the average latent energy during the DDIM sampling process can be expressed as follows:

$$
\begin{aligned}
\mathbb{E}[x_{t-1}^2] &= \mathbb{E}\left[\sqrt{\frac{\bar{\alpha}_{t-1}}{\bar{\alpha}_t}} x_t \right]^2 + \mathbb{E}\left[\left(\sqrt{1 - \bar{\alpha}_{t-1}} - \frac{\sqrt{\bar{\alpha}_{t-1}}\sqrt{1 - \bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}} \right) \tilde{\varepsilon}(x_t, t) \right]^2 \\
&+ 2 \times \mathbb{E}\left[\sqrt{\frac{\bar{\alpha}_{t-1}}{\bar{\alpha}_t}} x_t \right] \times \mathbb{E}\left[\left(\sqrt{1 - \bar{\alpha}_{t-1}} - \frac{\sqrt{\bar{\alpha}_{t-1}}\sqrt{1 - \bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}} \right) \tilde{\varepsilon}(x_t, t) \right].
\end{aligned}
\tag{11}
$$

We assume that the predicted noise $\tilde{\epsilon}$ follows a standard normal distribution, such that $\mathbb{E}[\tilde{\epsilon}(x_t, t)] = 0$. Under this assumption, the average latent energy of the DDIM sampler can be simplified as:

$$
\mathbb{E}[x_{t-1}^2] = \mathbb{E}\left[\sqrt{\frac{\bar{\alpha}_{t-1}}{\bar{\alpha}_t}} x_t \right]^2 + \mathbb{E}\left[\left(\sqrt{1 - \bar{\alpha}_{t-1}} - \frac{\sqrt{\bar{\alpha}_{t-1}}\sqrt{1 - \bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}} \right) \tilde{\varepsilon}(x_t, t) \right]^2.
\tag{12}
$$

Several previous works [20, 22, 57, 61] define the Signal-to-Noise Ratio (SNR) at a given timestep of a diffusion model as follows:

$$SNR_t = \frac{\bar{\alpha}_t}{1 - \bar{\alpha}_t}. \tag{13}$$

Several works [20, 22, 57, 61] have observed that the SNR must be adjusted during the generation process at different resolutions. Suppose the diffusion model is originally designed for a resolution of $H \times W$, and we aim to extend it to generate images at a higher resolution of $H' \times W'$, where $H' > H$ and $W' > W$. According to the derivations in [57, 61], the adjusted formulation of α_t is given as follows:

$$\bar{\alpha}'_t = \frac{\bar{\alpha}_t}{\gamma - (\gamma - 1)\bar{\alpha}_t}. \tag{14}$$

Here, the value of γ is typically defined as $(H'/H \cdot W'/W)^2$. By substituting the modified $\bar{\alpha}'_t$ into Eq. 10, we obtain the SNR-corrected sampling formulation as follows:

$$
\begin{aligned}
\mathbb{E}[x_{t-1}] &= \sqrt{\frac{\bar{\alpha}'_{t-1}}{\bar{\alpha}'_t}}\mathbb{E}[x_t] + \left(\sqrt{1 - \bar{\alpha}'_{t-1}} - \frac{\sqrt{\bar{\alpha}'_{t-1}}\sqrt{1 - \bar{\alpha}'_t}}{\sqrt{\bar{\alpha}'_t}}\right)\mathbb{E}[\tilde{\epsilon}(x_t, t)] \\
&= \sqrt{\frac{\frac{\bar{\alpha}_{t-1}}{\gamma - (\gamma - 1)\bar{\alpha}_{t-1}}}{\frac{\bar{\alpha}_t}{\gamma - (\gamma - 1)\bar{\alpha}_t}}}\mathbb{E}[x_t] + \left(\sqrt{1 - \frac{\bar{\alpha}_{t-1}}{\gamma - (\gamma - 1)\bar{\alpha}_{t-1}}} - \sqrt{\frac{\frac{\bar{\alpha}_{t-1}}{\gamma - (\gamma - 1)\bar{\alpha}_{t-1}}\left(1 - \frac{\bar{\alpha}_t}{\gamma - (\gamma - 1)\bar{\alpha}_t}\right)}{\frac{\bar{\alpha}_t}{\gamma - (\gamma - 1)\bar{\alpha}_t}}}\right)\mathbb{E}[\tilde{\epsilon}(x_t, t)] \\
&= \sqrt{\frac{\gamma - (\gamma - 1)\bar{\alpha}_t}{\gamma - (\gamma - 1)\bar{\alpha}_{t-1}}}\sqrt{\frac{\bar{\alpha}_{t-1}}{\bar{\alpha}_t}}\mathbb{E}[x_t] + \sqrt{\frac{\gamma}{\gamma - (\gamma - 1)\bar{\alpha}_{t-1}}}\left(\sqrt{1 - \bar{\alpha}_{t-1}} - \frac{\sqrt{\bar{\alpha}_{t-1}}\sqrt{1 - \bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}}\right)\mathbb{E}[\tilde{\epsilon}(x_t, t)].
\end{aligned} \tag{15}
$$

The average latent energy under SNR correction can be derived as follows:

$$
\begin{aligned}
\mathbb{E}[x_{t-1}^2] &= \mathbb{E}\left[\sqrt{\frac{\bar{\alpha}'_{t-1}}{\bar{\alpha}'_t}}x_t\right]^2 + \mathbb{E}\left[\left(\sqrt{1 - \bar{\alpha}'_{t-1}} - \frac{\sqrt{\bar{\alpha}'_{t-1}}\sqrt{1 - \bar{\alpha}'_t}}{\sqrt{\bar{\alpha}'_t}}\right)\tilde{\epsilon}(x_t, t)\right]^2 \\
&= \frac{\gamma - (\gamma - 1)\bar{\alpha}_t}{\gamma - (\gamma - 1)\bar{\alpha}_{t-1}}\mathbb{E}\left[\sqrt{\frac{\bar{\alpha}_{t-1}}{\bar{\alpha}_t}}x_t\right]^2 + \frac{\gamma}{\gamma - (\gamma - 1)\bar{\alpha}_{t-1}}\mathbb{E}\left[\left(\sqrt{1 - \bar{\alpha}_{t-1}} - \frac{\sqrt{\bar{\alpha}_{t-1}}\sqrt{1 - \bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}}\right)\tilde{\epsilon}(x_t, t)\right]^2.
\end{aligned} \tag{16}
$$

Compared to the original energy formulation Eqa. 12, two additional coefficients appear: $\frac{\gamma - (\gamma - 1)\bar{\alpha}_t}{\gamma - (\gamma - 1)\bar{\alpha}_{t-1}}$ and $\frac{\gamma}{\gamma - (\gamma - 1)\bar{\alpha}_{t-1}}$. Since $\bar{\alpha}_{t-1}$ and $\bar{\alpha}_t$ are very close, the first coefficient is approximately equal to 1. In the DDIM sampling formulation, $\bar{\alpha}_t$ is within the range [0, 1], which implies that the second coefficient falls within $[1, \gamma]$. As a result, after the SNR correction, the average latent energy increases. Therefore, SNR correction essentially serves as a mechanism for energy enhancement. In this sense, both energy rectification and SNR correction aim to increase the average latent energy. However, since our method allows for the flexible selection of hyperparameters, it can achieve superior performance.

7.3. Applying *RectifiedHR* to Stable Diffusion 3

To validate the effectiveness of our method on a transformer-based diffusion model, we apply it to `stable-diffusion-3-medium` using the `diffusers` library. As shown in Fig. 10, we compare the qualitative results of our method with those of direct inference at a resolution of 2048×2048. It can be observed that direct inference introduces grid artifacts and object deformations, whereas our method partially mitigates and corrects these issues.

15

Figure 10. Qualitative comparison on Stable Diffusion 3 at 2048×2048 resolution. The green boxes indicate enlarged views of local regions within the high-resolution image.

7.4. Ablation results on hyperparameters

In this section, we conduct ablation experiments on the hyperparameters in Eq. 7 and Eq. 9 of the main text using SDXL. The baseline hyperparameter settings follow those described in the Sec. 4.1. We vary one hyperparameter at a time while keeping the others fixed at the two target resolutions to evaluate the impact of each parameter on performance, as defined in Eq. 7 and Eq. 9 of the main text. The evaluation procedure for FID_c, FID_r, IS_c, and IS_r follows the protocol outlined in Sec. 7.7. All experiments are conducted on two NVIDIA A800 GPUs unless otherwise specified. As a result, the performance may differ slightly from experiments conducted using eight NVIDIA A800 GPUs.

In Eq. 7 and Eq. 9 of the main text, $\omega_{\min}$ and $T_{\max}$ are fixed and do not require ablation. The value of N in both equations is kept consistent. For the 2048×2048 resolution scene, with N set to 2, variations in M_T and M_ω do not significantly affect the results. Thus, only N, $\omega_{\max}$, and $T_{\min}$ are ablated. The quantitative ablation results for the 2048×2048 resolution are shown in Fig. 11, Fig. 12, and Fig. 13. For the 4096×4096 resolution scene, N, $\omega_{\max}$, $T_{\min}$, M_T, and M_ω are ablated. The corresponding quantitative ablation results for the 4096×4096 resolution are presented in Fig. 14, Fig. 15, Fig. 16, Fig. 17, and Fig. 18. Based on these results, it can be concluded that the basic numerical settings used in this experiment represent the optimal solution.

In Eq. 7 and Eq. 9 of the main text, $\omega_{\min}$ and $T_{\max}$ are fixed and thus excluded from ablation. The value of N is kept consistent across both equations. For the 2048×2048 resolution setting, with N set to 2, variations in M_T and M_ω have minimal impact on performance. Therefore, only N, $\omega_{\max}$, and $T_{\min}$ are subject to ablation. The corresponding quantitative ablation results are shown in Fig. 11, Fig. 12, and Fig. 13. For the 4096×4096 resolution setting, we ablate N, $\omega_{\max}$, $T_{\min}$, M_T, and M_ω. The corresponding results are presented in Fig. 14, Fig. 15, Fig.16, Fig.17, and Fig. 18. Based on these findings, we conclude that the default numerical settings used in our experiments yield the optimal performance.
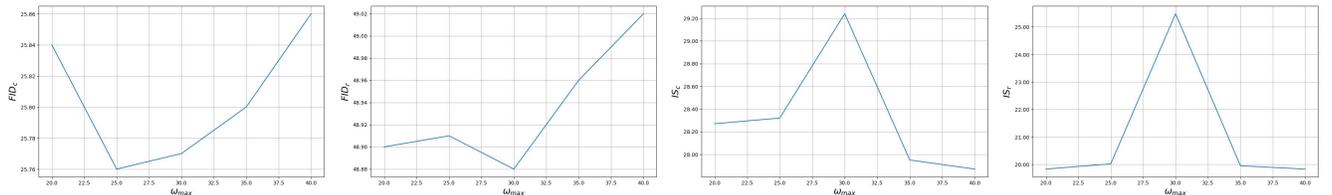

Figure 11. The image illustrates the ablation study of $\omega_{\max}$ in Eq. 9 of the main text for the 2048×2048 resolution setting. The values of $\omega_{\max}$ range over $20, 25, 30, 35, 40$.

7.5. Hyperparameter details and quantitative results for applying *RectifiedHR* to applications

The combination of *RectifiedHR* and WAN. *RectifiedHR* can be directly applied to video diffusion models such as WAN [56]. The officially supported maximum resolution for WAN 1.3B is 480×832 over 81 frames. Our goal is to generate videos at 960×1664 resolution using WAN 1.3B on an NVIDIA A800 GPU. The direct inference baseline refers to generating a 960×1664 resolution video directly using WAN 1.3B. In contrast, *WAN+RectifiedHR* refers to using *RectifiedHR* to generate the same-resolution video. The selected hyperparameters in Eq. 7 and Eq. 9 of the main text are: $N = 2$, $\omega_{\max} = 10$, $\omega_{\min} = 5$, $T_{\min} = 30$, $T_{\max} = 50$, $M_T = 1$, and $M_\omega = 1$.
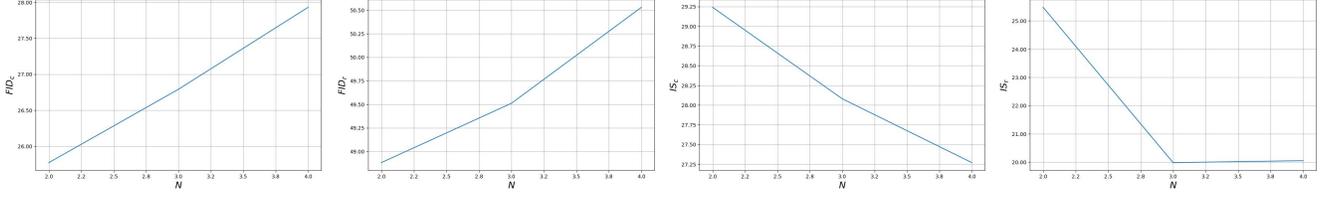
16

Figure 12. The image illustrates the ablation study of N in Eq. 7 and Eq. 9 of the main text for the 2048×2048 resolution setting. The values of N range over $2, 3, 4$.
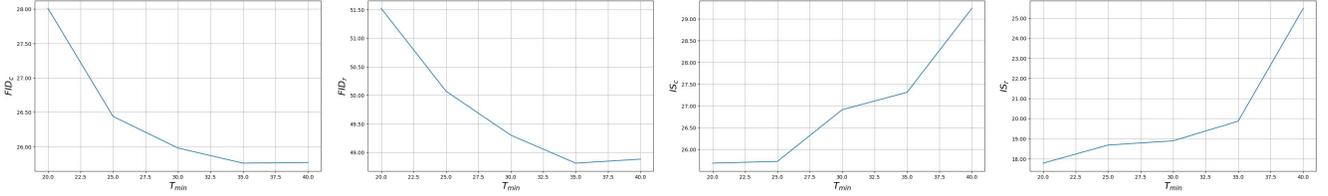

Figure 13. The image illustrates the ablation study of $T_{\min}$ in Eq. 7 of the main text for the 2048×2048 resolution setting. The values of $T_{\min}$ range over $20, 25, 30, 35, 40$.
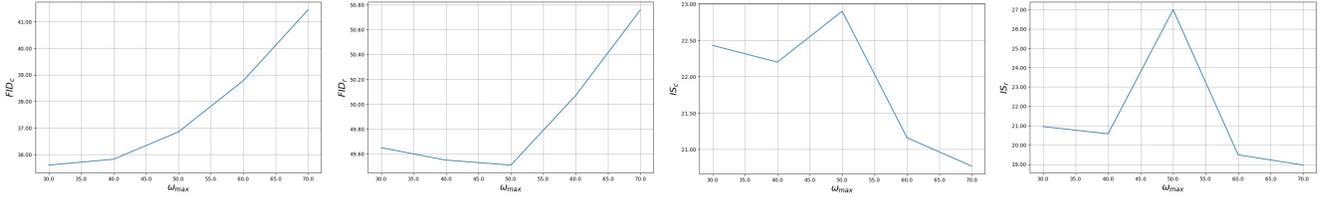

Figure 14. The image illustrates the ablation study of $\omega_{\max}$ in Eq. 9 of the main text for the 4096×4096 resolution setting. The values of $\omega_{\max}$ range over $30, 40, 50, 60, 70$.

Figure 15. The image illustrates the ablation study of M_ω in Eq. 9 of the main text for the 4096×4096 resolution setting. The values of M_ω range over $0.5, 1, 2$.
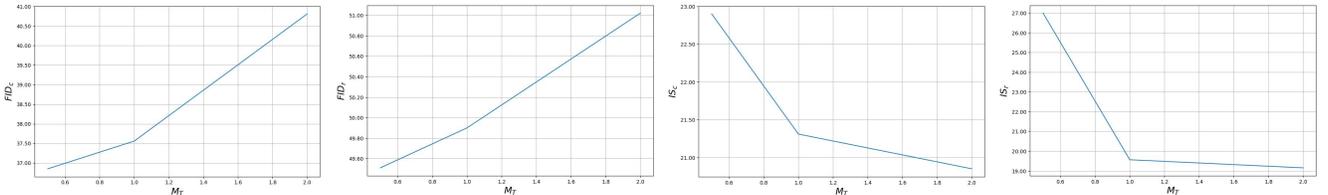

Figure 16. The image illustrates the ablation study of M_T in Eq. 7 of the main text for the 4096×4096 resolution setting. The values of M_T range over $0.5, 1, 2$.

The combination of *RectifiedHR* and OIR. *RectifiedHR* can also be applied to image editing tasks. We employ SDXL as the base model and randomly select several high-resolution images from the OIR-Bench [58] dataset for qualitative comparison. Specifically, we compare two approaches: (1) direct single-object editing using OIR [58], and (2) OIR combined with *RectifiedHR*. While the OIR baseline directly edits high-resolution images, the combined method first downsamples the input
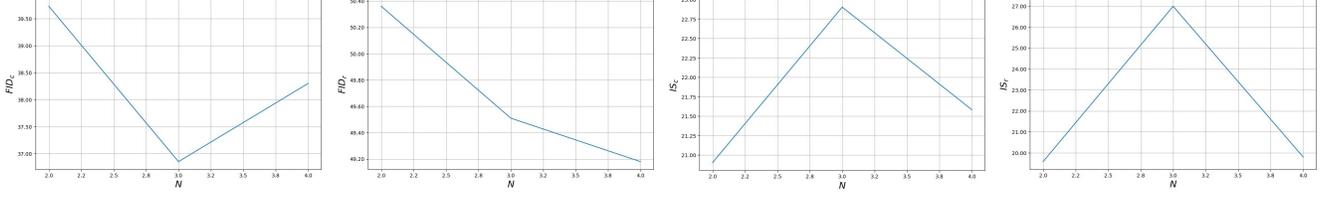
Figure 17. The image illustrates the ablation study of N in Eq. 7 and Eq. 9 of the main text for the 4096×4096 resolution setting. The values of N range over $2, 3, 4$.
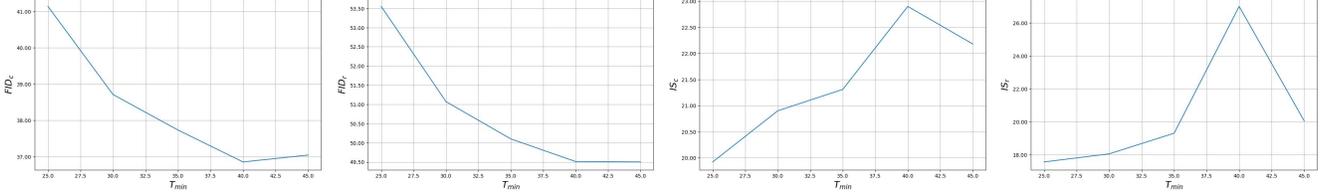

Figure 18. The image illustrates the ablation study of $T_{\min}$ in Eq. 7 of the main text for the 4096×4096 resolution setting. The values of $T_{\min}$ range over $25, 30, 35, 40, 45$.

to 1024×1024, performs editing via the OIR pipeline, and then applies *RectifiedHR* during the denoising phase to restore fine-grained image details. For the 2048×2048 resolution setting, the hyperparameters in Eq. 7 and Eq. 9 of the main text are: $N = 2$, $\omega_{\max} = 30$, $\omega_{\min} = 5$, $T_{\min} = 40$, $T_{\max} = 50$, $M_T = 1$, and $M_\omega = 1$. For the 3072×3072 resolution setting, the hyperparameters are: $N = 3$, $\omega_{\max} = 40$, $\omega_{\min} = 5$, $T_{\min} = 40$, $T_{\max} = 50$, $M_T = 1$, and $M_\omega = 1$.

The combination of *RectifiedHR* and DreamBooth. *RectifiedHR* can be directly adapted to various customization methods, where it is seamlessly integrated into DreamBooth without modifying any of the training logic of DreamBooth [46]. The base model for the experiment is SD1.4, which supports a native resolution of 512×512 and a target resolution of 1536×1536. The hyperparameters selected in Eq. 7 and Eq. 9 of the main text are as follows: N is 3, $\omega_{\max}$ is 30, $\omega_{\min}$ is 5, $T_{\min}$ is 40, $T_{\max}$ is 50, M_T is 1, and M_ω is 1. Furthermore, as demonstrated in Tab. 3, we conduct a quantitative comparison between the *RectifiedHR* and direct inference, using the DreamBooth dataset for testing. The test metrics and process were fully aligned with the methodology in [46]. It can be observed that *RectifiedHR* outperforms direct inference in terms of quantitative metrics for high-resolution customization generation.

RectifiedHR can be directly adapted to various customization methods and is seamlessly integrated into DreamBooth [46] without modifying any part of its training logic. The base model used in this experiment is SD1.4, which natively supports a resolution of 512×512, with the target resolution set to 1536×1536. The selected hyperparameters in Eq. 7 and Eq. 9 of the main text are as follows: $N = 3$, $\omega_{\max} = 30$, $\omega_{\min} = 5$, $T_{\min} = 40$, $T_{\max} = 50$, $M_T = 1$, and $M_\omega = 1$. Furthermore, as shown in Tab. 3, we conduct a quantitative comparison between *RectifiedHR* and direct inference using the DreamBooth dataset for evaluation. The test metrics and protocol are fully aligned with the methodology described in [46]. The results demonstrate that *RectifiedHR* outperforms direct inference in terms of quantitative metrics for high-resolution customization generation.

Direct Inference	DINO ↑	CLIP-I ↑	CLIP-T ↑
DreamBooth + RectifiedHR	**0.625**	**0.761**	**0.249**
DreamBooth	0.400	0.673	0.220

Table 3. Quantitative comparison results between *RectifiedHR* and direct inference after DreamBooth training. The evaluation is conducted on a scene with a resolution of 1536×1536.

The combination of *RectifiedHR* and ControlNet. Our method can be seamlessly integrated with ControlNet [59] to operate directly during the inference stage, enabling image generation conditioned on various control signals while simultaneously enhancing its ability to produce high-resolution outputs. The base model used is SDXL. The selected hyperparameters in Eq. 7 and Eq. 9 of the main text are: $N = 3$, $\omega_{\max} = 40$, $\omega_{\min} = 5$, $T_{\min} = 40$, $T_{\max} = 50$, $M_T = 1$, and $M_\omega = 1$.
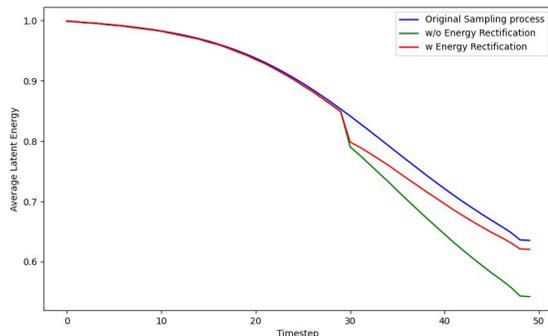
Figure 19. Visualization of the average latent energy curve following energy rectification.

7.6. Visualization of the energy rectification curve

To better visualize the average latent energy during the energy rectification process, we plot the corrected energy curves. We randomly select 100 prompts from LAION-5B for the experiments. As shown in Fig. 19, the blue line represents the energy curve at a resolution of 1024×1024. For the 2048×2048 resolution setting, we use the following hyperparameters: $T_{\min} = 30$, $T_{\max} = 50$, $N = 2$, $\omega_{\min} = 5$, $\omega_{\max} = 30$, $M_T = 1$, and $M_\omega = 1$. The red line corresponds to our method with energy rectification for generating 2048×2048 resolution images, while the green line shows the result of our method without the energy rectification module. It can be observed that energy rectification effectively compensates for energy decay.

7.7. Implementation details

Although a limited number of samples may lead to lower values for metrics such as FID [17], we follow prior protocols and randomly select 1,000 prompts from LAION-5B [49] for text-to-image generation. Evaluations are conducted using 50 inference steps, empty negative prompts, and fixed random seeds.

We employ four widely used quantitative metrics: Fréchet Inception Distance (FID) [17], Kernel Inception Distance (KID) [3], Inception Score (IS) [48], and CLIP Score [43]. FID and KID are computed using `pytorch-fid`, while CLIP Score and IS are computed using `torchmetrics`. Specifically, FID_r, KID_r, and IS_r require resizing images to 299×299. However, such evaluation is not ideal for high-resolution image generation. Following prior works [10, 33], we randomly crop 10 patches of size 1024×1024 from each generated high-resolution image to compute FID_s, KID_c, and IS_c.