A EXPERIMENT DETAILS

A.1 MODEL LAYERS IN OUR METHOD

The U-Net of Stable Diffusion (SD) v1.5, SD 2.1, and SD XL 1.0 share a similar convolution layer layout. We explain which layer to use re-dilated or dispersed convolutions without a loss of generality. We follow the naming of layers in diffuers[1]. A list of convolution layers contained in a U-Net block is shown in Tab. 1. The attention projection layers and convolution shortcut layers will not use re-dilation or dispersion since the convolution kernel in these layers is 1×1. Note that the first and the last convolution in the U-Net (conv_in and conv_out) will not use our method since they do not contribute to generating image contents. Also, the spatial part of the text-to-video model we used shares the same architecture as the SD. Therefore, layers of the following mentioned are also the same as our video experiment.

Layer name	Exist in all blocks	Use our method
attentions.0.proj_in	✓	✗
attentions.0.proj_out	✓	✗
attentions.1.proj_in	✗	✗
attentions.1.proj_out	✗	✗
attentions.2.proj_in	✗	✗
attentions.2.proj_out	✗	✗
resnets.0.conv1	✓	✓
resnets.0.conv2	✓	✓
resnets.0.conv_shortcut	✗	✗
resnets.1.conv1	✓	✓
resnets.1.conv2	✓	✓
downsamplers.0.conv	✗	✓

Table 1: The layers to use our method in a U-Net block. The second column shows the existence condition since some layers cannot be seen in specific U-Net blocks.

A.2 HYPERPARAMETERS

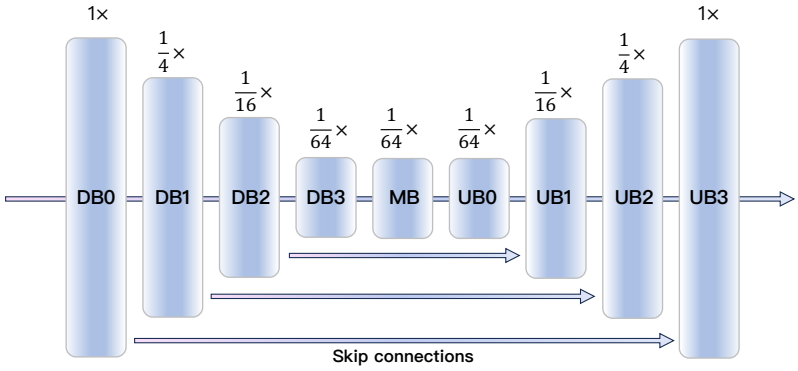

Figure 1: Reference block names in the following experiment details. The fractional multiples above blocks are the spatial pixel number of feature maps within the block compared to the network input. i.e, the input latent has 64^2 spatial dimension, then the size of feature maps in DB3 to UB0 is 8^2.

[1] https://github.com/huggingface/diffusers

We explain our selection for hyperparameters in this section. All samples are generated using the default classifier-free guidance scale of the corresponding pre-trained model (i.e. SD 1.5 and SD 2.1 use 7.5, SD XL 1.0 uses 5.0). Our SD 2.1 experiments use a similar setting to SD 1.5. We list the hyperparameters for SD 1.5 only for brevity. The evaluation settings for SD 1.5 are shown in Tab. 2, 3, 4, 5. The settings for SD XL 1.0 are shown in Fig. 6, 7, 8, 9. A reference for block names and their exact location in the U-Net can be found in Fig. 1. The tables show detailed settings about which block to use re-dilation conv and dispersed conv. Dilation scale rb. means the dilation scale for re-dilated blocks and dilation scale db. defines the dilation scale for dispersed blocks. If the sampling uses noise-damped classifier-free guidance, we construct a $\epsilon_\theta(\cdot)$ with strong denoising capability by turning some outskirt blocks that use re-dilated and dispersed convolution to the original blocks. The chosen ones that become the original blocks are listed in noise-damped blocks.

Params	Values
latent resolution	4×128×128
re-dilated blocks	[DB3, MB, UB0]
dilation scale rb.	[2, 2, 2]
dispersed blocks	∅
progressive	✗
noise-damped cfg.	✗
inference timesteps	50
τ	30

Table 2: 1024^2 SD 1.5 experiment settings.

Params	Values
latent resolution	4×160×160
re-dilated blocks	[DB3, MB, UB0]
dilation scale rb.	[2.5, 2.5, 2.5]
dispersed blocks	∅
progressive	✗
noise-damped cfg.	✗
inference timesteps	50
τ	30

Table 3: 1280^2 SD 1.5 experiment settings.

Params	Values
latent resolution	4×128×256
re-dilated blocks	[DB0, DB1, DB2, DB3, MB, UB0, UB1, UB2, UB3]
dilation scale rb.	[2, 2, 2, 2, 2, 2, 2, 2, 2]
dispersed blocks	∅
progressive	✗
noise-damped cfg.	✓
noise-damped blocks	[DB0, DB1, DB2, UB1, UB2, UB3]
inference timesteps	50
τ	30

Table 4: 2048×1024 SD 1.5 experiment settings.

Params	Values
latent resolution	4×256×256
re-dilated blocks	[DB0, DB1, UB2, UB3]
dilation scale rb.	[2, 4, 4, 2]
dispersed blocks	[DB2, DB3, MB, UB0, UB1]
dilation scale db.	[2, 2, 2, 2, 2]
dispersed kernel size	$3 \times 3 \rightarrow 5 \times 5$
progressive	✓
noise-damped cfg.	✓
noise-damped blocks	[DB0, DB1, UB2, UB3]
inference timesteps	50
τ	35

Table 5: 2048^2 SD 1.5 experiment settings.

A.3 SYNCHRONIZE STATISTICS BETWEEN TILES IN GROUPNORM

Params	Values
latent resolution	4×256×256
re-dilated blocks	[DB3, MB, UB0]
dilation scale rb.	[2, 2, 2]
dispersed blocks	∅
progressive	✗
noise-damped cfg.	✗
inference timesteps	50
τ	30

Table 6: 2048^2 SD XL 1.0 settings.

Params	Values
latent resolution	4×320×320
re-dilated blocks	[DB1, DB2, DB3, MB, UB0, UB1, UB2]
dilation scale rb.	[2, 2, 2.5, 2.5, 2.5, 2, 2]
dispersed blocks	∅
progressive	✗
noise-damped cfg.	✓
noise-damped blocks	[DB1, DB2, UB1, UB2]
inference timesteps	50
τ	30

Table 7: 2560^2 SD XL 1.0 experiment settings.

Params	Values
latent resolution	4×256×512
re-dilated blocks	[DB1, DB2, DB3, MB, UB0, UB1, UB2]
dilation scale rb.	[2, 2, 2, 2, 2, 2, 2]
dispersed blocks	∅
progressive	✗
noise-damped cfg.	✓
noise-damped blocks	[DB1, DB2, UB1, UB2]
inference timesteps	50
τ	30

Table 8: 4096×2048 SD XL 1.0 experiment settings.

Params	Values
latent resolution	4×512×512
re-dilated blocks	[DB2, UB1]
dilation scale rb.	[2, 2]
dispersed blocks	[DB3, MB, UB0]
dilation scale db.	[2, 2]
dispersed kernel size	$3 \times 3 \rightarrow 5 \times 5$
progressive	✓
noise-damped cfg.	✓
noise-damped blocks	[DB2, UB1]
inference timesteps	50
τ	35

Table 9: 4096^2 SD XL settings.

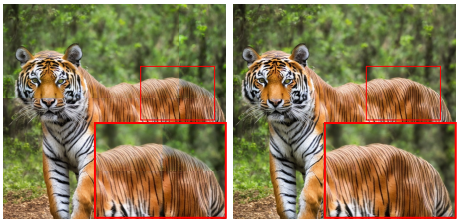

Figure 2: Direct tiled decode causes abrupt changes in tile borders and different color tones in tiles. We synchronize the statistics in VAE GroupNorm between tiles to address this problem.

When the generated image size is large (i.e., $> 2048 \times 2048$), the VAE of SD requires enormous VRAM for decoding and is usually not applicable on a personal GPU. A simple solution is decoding in tiles. However, tiled decoding usually causes abrupt changes between different tiles as shown in Fig. 2. To solve this, one can make overlapped regions between tiles and interpolate on the overlapped regions. However, another problem of tiled decoding is the inconsistent color tone between tiles. We figure out this is caused by the independent computation of GroupNorm (GN) layers in VAE between tiles. We propose to synchronize the feature statistics in GN in different tiles. Specifically, we compute the mean and std using all tiles instead of using only current ones. As shown in Fig. 2, it eliminates the color tone difference efficiently.

B RE-DILATED ATTENTION

Figure 3: Illustration and results of two re-dilations.

Here, we introduce the experimented re-dilated attention. We aim to keep the original receptive field of attention, e.g., the attention token quantity. Thus, before calculating the attentional features, we first split the input feature map into four slices (the resolution is 4x higher than the training), and for each slice, we flat them into token sequences and feed them into the QKV attention. After the attention calculation, we merge them back to form the original feature arrangement. This operation strictly controls the token length of attention to be the same as training. However, this cannot solve the structure issue of the generated image, as shown in the 2^{nd} row of Fig. 3. However, when applying the redilation on the convolutional kernel, the structure is totally correct. This demonstrates that the key cause of structure repetition lies in convolutional kernels.

C More Comparisons

Method	FID_r	KID_r	$pFID_r$	$pKID_r$	$sFID_r$	$sKID_r$	Time (s)	#param
SD XL	18.50	0.005	29.63	0.014	16.68	0.007	6.5	3.5B
SD 2.1+SR	15.39	0.005	**17.30**	**0.005**	14.57	0.007	9.5 (1.5+8)	2.2B
Ours (SD 2.1)	18.73	0.005	20.97	**0.005**	**10.17**	**0.004**	**5.6**	**1.3B**
Ours (SD 2.1)+ LR	**9.96**	**0.003**	19.27	0.007	11.05	**0.004**	6.3(1.5+4.8)	**1.3B**

Table 10: Comparison results with state-of-the-art image generation models and super-resolution (SR) model under the resolution of 1024^2. Time indicates the second used for synthesizing one image on one A100 GPU with 16-bit precision). #param stands for the number of model parameters. $pFID_r$ and $pKID_r$ represent patch-FID and patch-KID, respectively. $pFID_r$ Chai et al. (2022), $pKID_r$ Chai et al. (2022), $sFID_r$ Nash et al. (2021), and $sKID_r$ Nash et al. (2021) are used to measure the texture details of generated samples.

We make a comprehensive comparison regarding general generation quality (FID_r, KID_r), texture details ($pFID_r$, $pKID_r$, $sFID_r$, $sKID_r$), inference time and number of model parameters with SD+SR and high-resolution image generation method SD XL, as shown in Tab. 10. Specifically, pFID/pKID avoids the downsampling operation and instead uses cropping in the metric calculation. sFID/sKID uses the features before the global average pooling to retain low-level details in the feature for the metric calculation, as well as avoids downsampling. The evaluation dataset is a 30k subset from Laion-5B with a resolution larger than 1024^2.

Results show that our training-free method (with no low-resolution reference image) achieves almost the comparable generation performance compared with well-trained SD+SR. Additionally, we achieve better texture details than SD + SR (see the sFID and sKID metrics, as well as the user study in the main paper). With the low-resolution generated samples as guidance, our method achieves much better results than SD+SR. At the same time, our method has 59% inference time and 59% model parameters less than SD+SR, showing our better efficiency. Compared with SD XL, we achieve both better metrics and lower inference time and parameter numbers.

D More ablations

We conduct additional ablation experiments to investigate the impact of increasing inference resolutions. As depicted in Figure 4, it is evident that as the inference resolution increases, the degree of structure distortion becomes more pronounced. Despite these challenges, our method is capable of effectively addressing and mitigating these issues, even in extremely demanding settings.

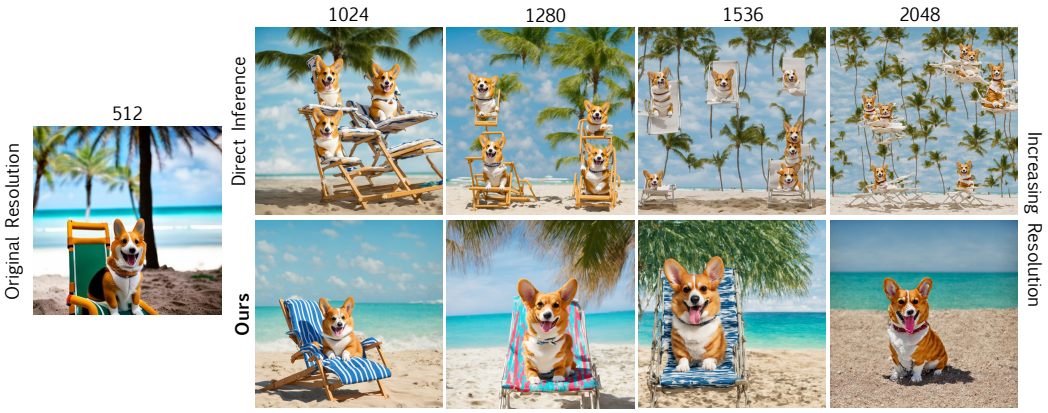

Figure 4: Performance change when increasing the image resolution. When increasing the resolution, the problem becomes more challenging. Our method is still capable of addressing these issues and maintaining a correct image structure.

E LIMITATIONS

Ensuring the accuracy of the local structure remains challenging, as demonstrated in Figure 5, particularly with regard to intricate details like the fingers of the robot and the legs of the chairs. It is worth noting that this issue is not exclusive to our method but also exists in the original lower-resolution model. As our approach is training-free, it inherits the limitations of the original model.

Figure 5: Failure cases on local structures. Our method has failures in the local structure of the generated image. We observed that the original lower-resolution model also struggled with this problem.

F CLOSED-FROM SOLUTION FOR DISPERSED CONVOLUTION

We follow the notation in our main paper. Given a convolution layer with kernel $\boldsymbol{k} \in \mathbb{R}^{r\times r}$ and a target kernel size r'. We find a dispersion transform $\mathbf{R} \in \mathbb{R}^{r'^2\times r^2}$ to get a dispersed kernel $\boldsymbol{k}'$. Considering an input feature map $\boldsymbol{h} \in \mathbb{R}^{n\times n}$, we ignore the bias in convolution, since it will not influence our results. Then structure-level calibration and pixel-level calibration can form an equation set:

$$\begin{aligned} \text{interp}_d(f_k(\boldsymbol{h})) &= f_{\boldsymbol{k}'}(\text{interp}_d(\boldsymbol{h})) \\ \eta f_{\boldsymbol{k}}(\boldsymbol{h}) &= \eta f_{\boldsymbol{k}'}(\boldsymbol{h}), \end{aligned} \tag{1}$$

where $\text{interp}_d(f_k(\boldsymbol{h})) \in \mathbb{R}^{nd\times nd}$, $f_{\boldsymbol{k}}(\boldsymbol{h}) \in \mathbb{R}^{n\times n}$. The equation set has $(nd)^2 + n^2$ equations. Each equation is made up of the sum of terms $k_{ij}h_{ij}$ (elements in $\boldsymbol{k}$ and $\boldsymbol{h}$) where the coefficient of the terms are linear combinations of R_{ij} (elements in $\mathbf{R}$) and constants. For example, when $n = 3, r = 3$ and $r' = 5$, the 5-th equation in pixel-level calibration is:

$$\begin{aligned} &k_{11}h_{11} + k_{12}h_{12} + k_{13}h_{13} + \cdots + k_{33}h_{33} = \\ &(R_{7,1}k_{11} + R_{7,2}k_{12} + R_{7,3}k_{13} + \cdots + R_{7,9}k_{33})h_{11} + \\ &(R_{8,1}k_{11} + R_{8,2}k_{12} + R_{8,3}k_{13} + \cdots + R_{8,9}k_{33})h_{12} + \\ &\cdots + \\ &(R_{19,1}k_{11} + R_{19,2}k_{12} + R_{19,3}k_{13} + \cdots + R_{19,9}k_{33})h_{33} \end{aligned} \tag{2}$$

We rearrange the equation and get:

$$(1 - R_{7,1})k_{11}h_{11} + (-R_{7,2})k_{12}h_{11} + \cdots (1 - R_{19,9})k_{33}h_{33} = 0. \tag{3}$$

To ensure this equation for all $k_{ij}h_{ij}$, one can let every coefficient be zero. Getting a linear equation set for the 5-th equation in pixel-level calibration.

$$\begin{cases} R_{7,1} = -1 \\ R_{7,2} = 0 \\ \quad \cdots \\ R_{19,9} = -1 \end{cases} \tag{4}$$

We do this for every equation in Eq. (1) to derive a larger set of linear equations of R_{ij}. Note that the equation sets are linear equations of R_{ij} and have no $\boldsymbol{h}$ or $\boldsymbol{k}$, making it applicable to all conv kernels and any input feature. Let's go back to the general case where $\mathbf{R} \in \mathbb{R}^{r'^2 \times r^2}$. Let $A_{\text{structure}}$ denote the coefficient of the linear combination for R_{ij} in structure-level calibration, and let $b_{\text{structure}}$ denote the right-hand-side constants in the equation set of structure-level calibration. Similarly, we define $A_{\text{pixel}}, b_{\text{pixel}}$, and we construct a least square problem.

$$A\mathbf{x} = b, \quad A = \begin{bmatrix} A_{\text{structure}} \\ \eta A_{\text{pixel}} \end{bmatrix}, b = \begin{bmatrix} b_{\text{structure}} \\ \eta b_{\text{pixel}} \end{bmatrix}, \mathbf{x} = \begin{bmatrix} R_{1,1} \\ R_{1,2} \\ \cdots \\ R_{r'^2,r^2} \end{bmatrix}. \tag{5}$$

The solution is $\mathbf{x} = (A^T A)^{-1} A^T b$. This can be easily solved by math software, i.e., MATLAB.

G SEARCHING RE-DILATION SCHEDULE

We first test a series of hand-crafted hyperparameters and figure out some empirical results: 1) Using re-dilation/dispersion in inner U-Net blocks and using original convolution in outer U-Net blocks produces good results. 2) Sharing the same re-dilation scale in a block instead of using different re-dilation scales for every convolution layer within a block produces better results.

Then, we search for hyperparameters using an automatic strategy. For each target resolution, we sample 50 examples from LAION-5B to build an evaluation set. The MSE loss of noise estimation, used in the training of diffusion models, serves as a metric for the hyperparameter search at each diffusion timestep. The hyperparameter that achieves the lowest loss is chosen for the corresponding timestep.

To speed up the search process, we prune the hyperparameter set using the previously established empirical rules. The pruning method is as follows: 1) Use the same re-dilation scale in all convolution layers within a block. 2) The blocks within a start block and an end block will use re-dilation/dispersion. For example, SD v2.1 blocks are shown in Fig. 1. If the start block is DB2 and the end block is UB1, then the blocks that use re-dilation/dispersion are DB2, DB3, MB, UB0, and UB1. 3) If the search includes noise-damped classifier-free guidance, then the blocks that use re-dilation/dispersion in $\boldsymbol{\epsilon}_\theta$ is a continuous subset of the blocks that use re-dilation/dispersion in $\tilde{\boldsymbol{\epsilon}}_\theta$. For example, if $\tilde{\boldsymbol{\epsilon}}_\theta$ re-dilation/dispersion blocks are DB2, DB3, MB, UB0, UB1, that in $\boldsymbol{\epsilon}_\theta$ can be DB3, MB, UB0. 4) The maximum re-dilation scale does not exceed the enlargement scale of the target resolution (i.e., generating a 2048×2048 image using a 512×512 model, the maximum re-dilation scale is 4). The search step of the re-dilation scale is 1.

H OTHER VISUALIZATIONS

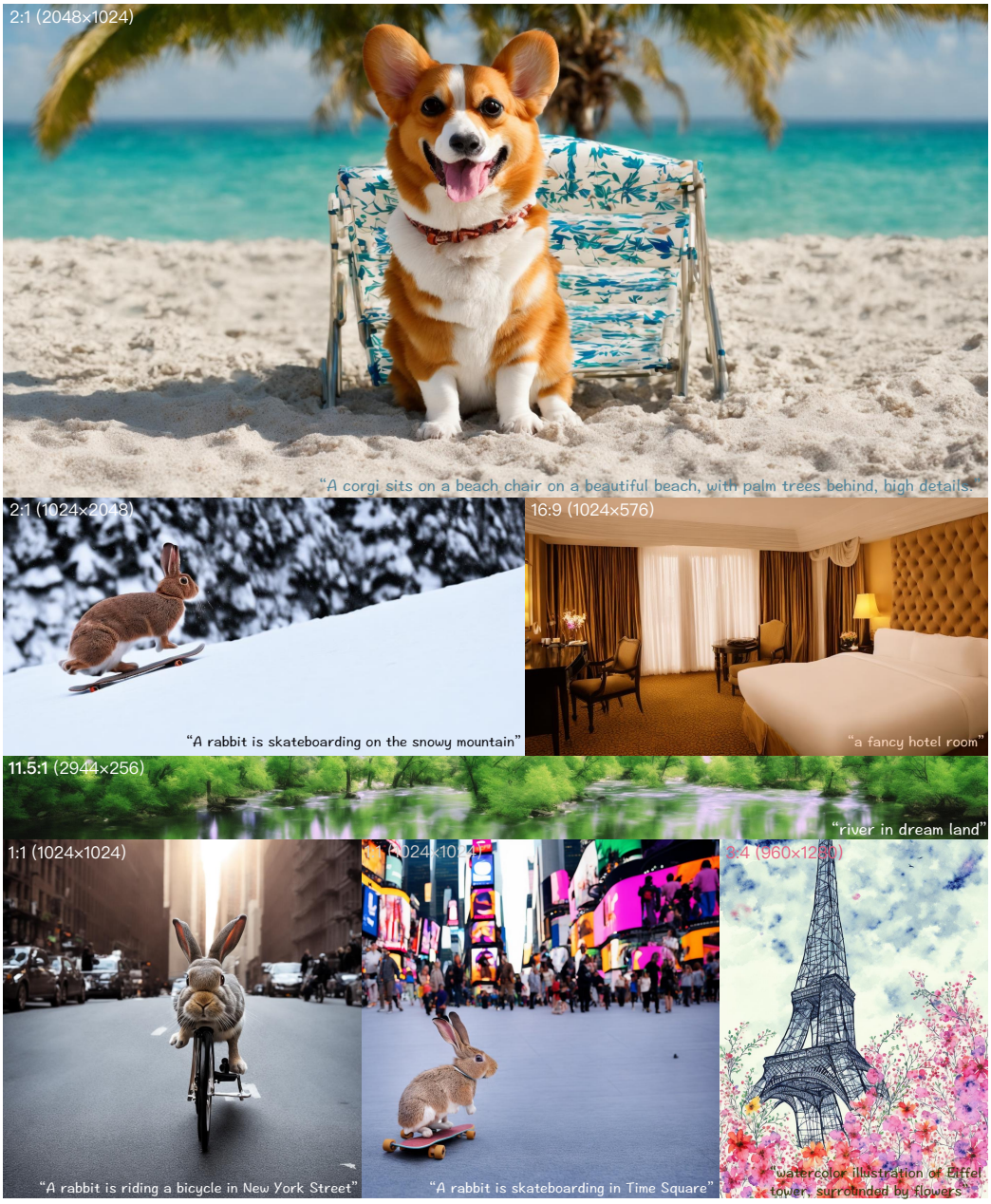

Figure 6: More generated results with our method and SD 2.1 with arbitrary aspect ratios and sizes.

Figure 7: More generated results with our method and SD 1.5 with arbitrary aspect ratios and sizes.

Figure 8: More generated results with our method and SD XL 1.0 with arbitrary aspect ratios and sizes.

REFERENCES

Lucy Chai, Michael Gharbi, Eli Shechtman, Phillip Isola, and Richard Zhang. Any-resolution training for high-resolution image synthesis. In *European Conference on Computer Vision*, pp. 170–188. Springer, 2022.

Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W Battaglia. Generating images with sparse representations. *arXiv preprint arXiv:2103.03841*, 2021.