Make a Cheap Scaling: A Self-Cascade Diffusion Model for Higher-Resolution Adaptation

Lanqing Guo[1,2†] Yingqing He[2,3†] Haoxin Chen[2] Menghan Xia[2] Xiaodong Cun[2] Yufei Wang[1]
Siyu Huang[4] Yong Zhang[2*] Xintao Wang[2] Qifeng Chen[3] Ying Shan[2] Bihan Wen[1*]
[1]Nanyang Technological University [2]Tencent AI Lab
[3]The Hong Kong University of Science and Technology [4]Clemson University

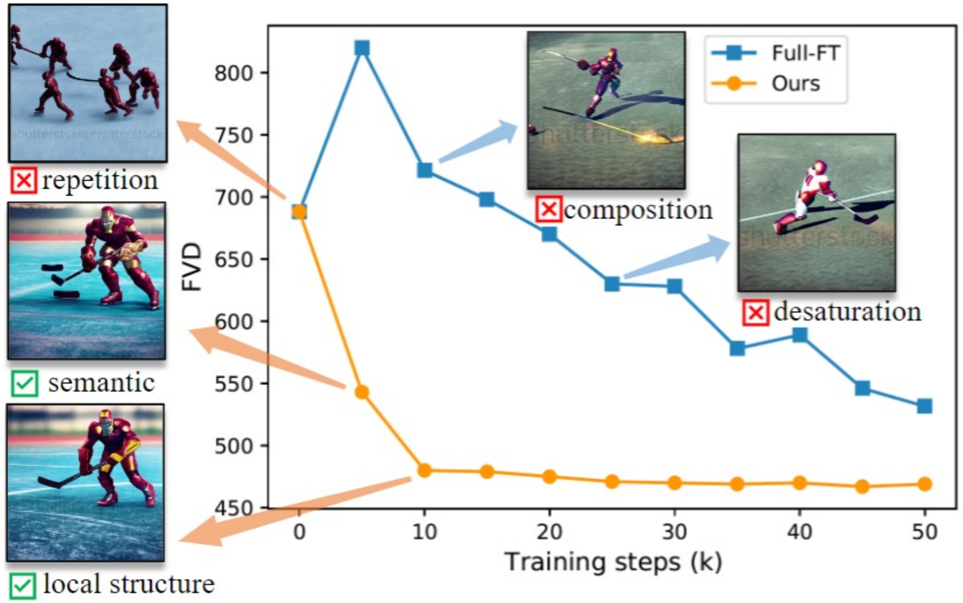
Motivation

For Stable Diffusion：

- 目前大部分Diffusion-based Model都是用单尺度(Single-Scale)的数据进行训练，生成比训练时分辨率大的图片时面临着挑战；
- A instance：使用在512×512的分辨率下Trained SD 2.1来生成1024×1024的图像会导致目标重复和构图能力下降的问题。

For Tuning for Stable Diffusion：

- 为了提高分辨率来Fine-tune Pretrained SD需要大量的计算资源
- A instance：SD 2.1在256×256分辨率下要训练550k steps，而在512×512分辨率下Fine-tune要 > 1000k steps。
- 如果Insufficient Tuning，还会导致去饱和和目标不合理的问题



Motivation

For Tuning-Free Methods：

- 需要对超参进行精确地调整；
- A instance：ScaleCrafter：使用Dilated Conv扩大感受野来适应新的分辨率图像生成

For Additional Parameters for Fine-tuning：

- 无法自适应于Scale的变化，且仍需大量的Fine-tuning steps (LORA)

For Cascaded Diffusion Model / Latent Diffusion Model：

- 级联的新的模型跟之前的模型参数并不共享，训练参数成倍增加
- 限制更高分辨率的尺度扩展能力

Goal：尺度自适应；Tuning-Free/Tuning-Efficient；不存在目标重复/构图差等问题

Preliminary

Stable Diffusion (SD)：在一个低维隐空间下进行扩散/去噪过程

包含有AutoEncoder、Decoder、Diffusion Model和Text-Conditional UNet Denoiser

AutoEncoder：$x_0 \in \mathbb{R}^{3\times H\times W} \rightarrow z_0 \in \mathbb{R}^{4\times H'\times W'}$：$z_0 = E(x_0)$

Decoder：reconstruct x_0 from z_0：$\hat{x}_0 = D(z_0) \approx x_0$

Diffusion Model：Fixed Diffusion Forward Process (Add noise)：

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I})$$

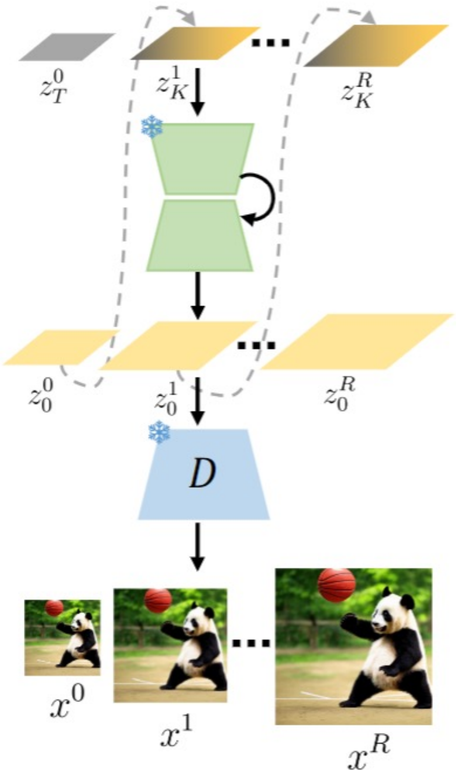
Denoiser：Predict Noise with Text Condition：

$$p_\theta(z_{0:T}|c) = \prod_{t=1}^{T} p_\theta(z_{t-1}|z_t, c)$$

Optimization： $\mathcal{L} = \mathbb{E}_{z_t,c,\epsilon,t}(\|\epsilon - \epsilon_\theta(z_t,t,c)\|^2)$ $\quad z_t = \sqrt{\bar{\alpha}_t}z_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \epsilon \in \mathcal{N}(0,\mathbf{I})$

Self-Cascade Diffusion Model

Goal：给定一个预训练的合成低分辨率图像的Stable Diffusion Model，目标是用一个自适应模型，在时间/资源/参数三者高效的方式生成更高分辨率的图像



Scale Decomposition

将目标的Scale d_R分解为多个逐步递增的Scale：

$$d = d_0 < d_1 \ldots < d_R \quad R = \lceil \log_4 d_R / d \rceil$$

每阶段Scale的生成结果会作用于下一阶段的Scale下的生成：

$$p_\theta(z_{0:T}^r | c, z^{r-1}) = p(z_T^r) \prod_{t=1}^{T} p_\theta(z_{t-1}^r | z_t^r, c, z^{r-1})$$

如：在Scale r下的去噪过程不仅跟condition c和timestep t有关，还跟Scale $r-1$下的Clean Result z^{r-1}有关

Self-Cascade Diffusion Model

Pivot-Guided Noise Re-schedule (Tuning-Free)

建立于一个观察/假设：

z^r和z^{r-1}在扩散过程的中间步中信息容量差距不显著，故假设 $p(z_K^r|z_0^{r-1})$ 为 $p(z_K^r|z_0^r)$ 的Proxy

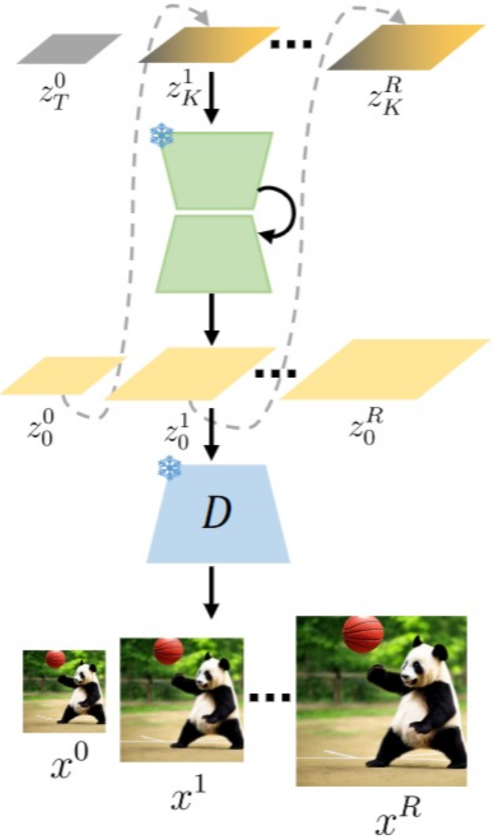
具体地，假设 ϕ_r 代表一个确定的resize插值函数，upsample $d_{r-1} \rightarrow d_r$；

然后把z_0^{r-1} upsample为$\phi_r(z_0^{r-1})$，然后Diffuse生成z_K^{r-1}来代替z_K^r：

$$z_K^r \sim \mathcal{N}(\sqrt{\bar{\alpha}_K}\phi_r(z_0^{r-1}), \sqrt{1-\bar{\alpha}_K}\mathbf{I})$$

最后denoising $K \rightarrow 0$，把结果应用于下一个Scale：

$$\begin{array}{l}
z_T^0 \rightarrow \cdots \rightarrow z_K^0 \rightleftharpoons \cdots \rightleftharpoons z_1^0 \rightleftharpoons z_0^0 \\
\qquad\qquad\quad \downarrow \\
\qquad\qquad\quad z_K^1 \rightleftharpoons \cdots \rightleftharpoons z_1^1 \rightleftharpoons z_0^1 \\
\qquad\qquad\quad \cdots \\
\qquad\qquad\quad \downarrow \\
\qquad\qquad\quad z_K^R \rightarrow \cdots \rightarrow z_1^R \rightarrow z_0^R
\end{array}$$



Self-Cascade Diffusion Model

Time-Aware Feature Upsampler (Tuning)

上述Tuning-Free的Limitation：因为Unseen higher-resolution GT，无法生成Detailed low-level structures

⇩

Tuning Self-Cascade Diffusion Model ⟶ Lightweight Time-Aware Feature Upsampler

使用一系列的Time-Aware的Feature Upsamplers $\Phi = \{\phi_1, \phi_2, \ldots, \phi_N\}$ ⇦ $N = 4$，0.002M Parameters

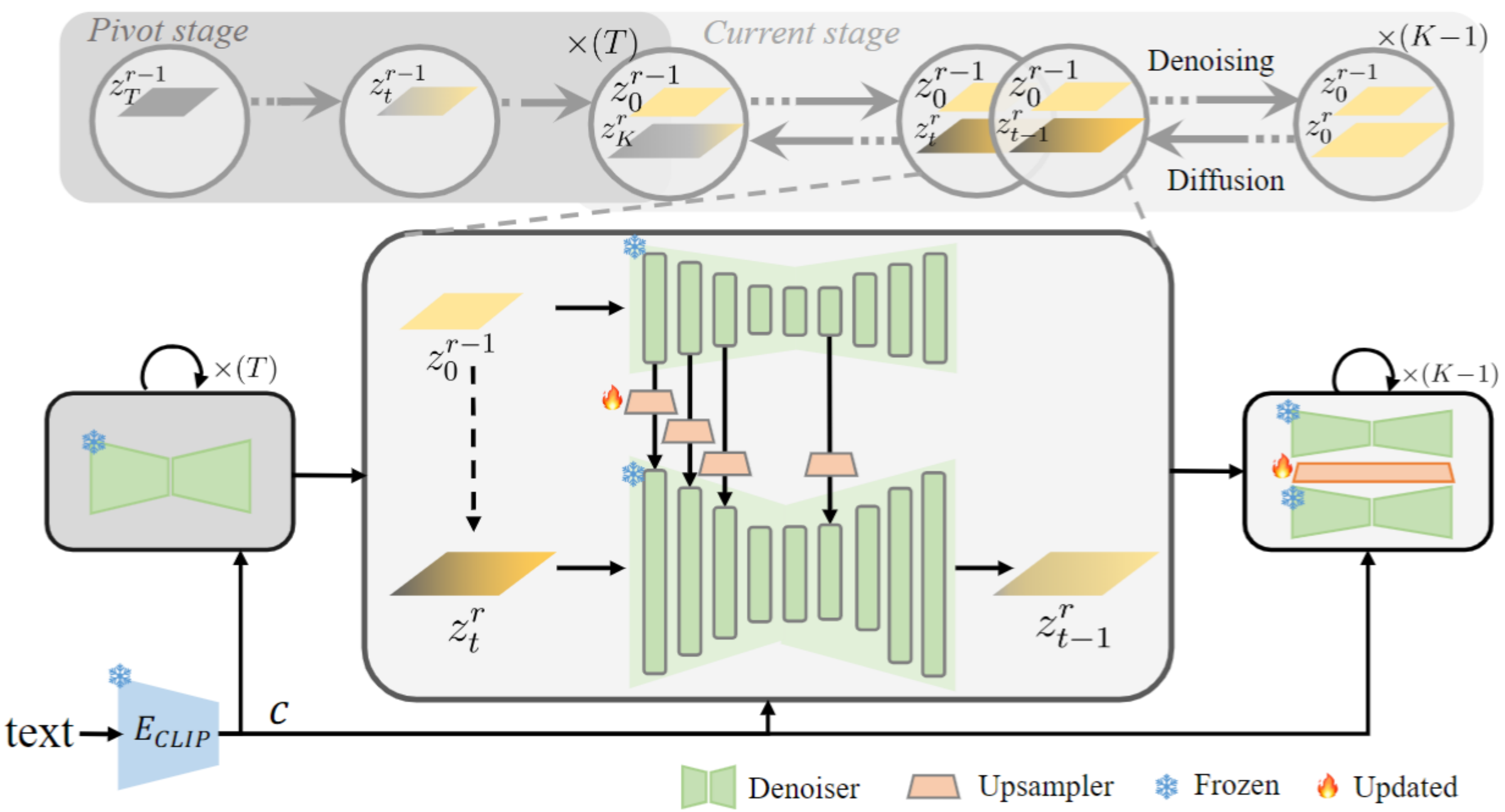
把pivot guidance z_0^{r-1}经过pre-trained UNet得到中间Skip Features嵌入到z_t^r下的UNet的Skip Features

- 选择Skip Features的原因：对生成的图像的质量影响忽略不计，仍可提供语义信息；
- Time-Aware的原因：随着噪声的消除，图像信噪比增加，去噪的焦点从高级语义结构转移到低级详细结构，Upsamplers需要适应不同的Timesteps

$$\hat{h}_{n,t}^r = h_{n,t}^r + \phi_n(h_{n,0}^{r-1}, t), \quad n \in \{1, \ldots, N\}$$

Self-Cascade Diffusion Model

Time-Aware Feature Upsampler (Tuning)



Algorithm 1 Time-aware feature upsampler tuning.

while not converged **do**
  $(z_0^r, z_0^{r-1}, c) \sim p(z^r, z^{r-1}, c)$
  $t \sim \text{Uniform}\{1, \dots, K\}$
  $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $z_t^r = \sqrt{\bar{\alpha}_t} z_0^r + \sqrt{1 - \bar{\alpha}_t} \epsilon$
  $\theta_\Phi \leftarrow \theta_\Phi - \eta \bigtriangledown_{\theta_\Phi} \|\tilde{\epsilon}_{\theta + \theta_\Phi}(z_t^r, t, c, z_0^{r-1}) - \epsilon\|^2$
end while
return θ_Φ

Algorithm 2 Pivot-guided inference for $\mathbb{R}^{d_{r-1}} \rightarrow \mathbb{R}^{d_r}$.

Input: text embedding c
if $r = 1$ **then**
  $z_T^r \sim \mathcal{N}(0, \mathbf{I})$
  for $t = T, \dots, 1$ **do**
    $z_{t-1}^r \sim p_\theta(z_{t-1}^r | z_t^r, c)$
  end for
else
  $z_K^r \sim q(z_K^r | z_0^{r-1})$
  for $t = K, \dots, 1$ **do**
    $z_{t-1}^r \sim p_\theta(z_{t-1}^r | z_t^r, c, z_0^{r-1})$
  end for
end if
return z_0^r

Experiments

Implement Details:

- Train on two A100 GPUs
- Diffusion Steps: 1000; DDIM Inference Steps: 50
- $N = 4, K = 700$

Evaluation Experiments:

1. T2I Models: SD 2.1 and SD XL 1.0 (Adapt to unseen higher-resolution domains)

Trained with 512×512, Inference Resolutions are 1024×1024 and 2048×2048

2. T2V Models: LVDM (Trained with $16\times256\times256$, Inference Resolutions are $16\times512\times512$)

Experiments

Image Generation

Methods	# Trainable Param	Training Step	Infer Time	$\text{FID}_r\downarrow$	$\text{KID}_r\downarrow$	$\text{FID}_b\downarrow$	$\text{KID}_b\downarrow$
Original	0	-	$1\times$	29.89	0.010	24.21	0.007
Attn-SF [16]	0	-	$1\times$	29.95	0.010	22.75	0.007
ScaleCrafter [10]	0	-	$1\times$	20.88	0.008	16.67	0.005
Ours-TF (Tuning-Free)	0	-	$1.04\times$	**12.25**	**0.004**	**6.09**	**0.001**
Full Fine-tuning ($18k$)	860M	$18k$	$1\times$	21.88	0.007	17.14	0.005
LORA-R32	15M	$18k$	$1.22\times$	17.02	0.005	11.33	0.003
LORA-R4	1.9M	$18k$	$1.20\times$	14.74	0.005	9.47	0.002
SD+SR	184M	1.25M	$5\times$	12.59	0.005	-	-
Ours-T (Tuning)	0.002M	$4k$	$1.06\times$	**12.40**	**0.004**	**3.15**	**0.0005**

Table 1. Quantitative results of different methods on the dataset of *Laion-5B* with $4\times$ adaptation on 1024^2 resolution. The best results are highlighted in **bold**. Note that Ours-TF and Ours-T denote the training-free version and the upsampler tuning version, respectively. # Param denotes the number of trainable parameters and Infer Time denotes the inference time of different methods *v.s.* original baseline. We put '-' since $\text{FID}_b/\text{KID}_b$ are unavailable for SD+SR [1].

Methods	$\text{FID}_r\downarrow$	$\text{KID}_r\downarrow$	$\text{FID}_b\downarrow$	$\text{KID}_b\downarrow$
Original	104.70	0.043	104.10	0.040
Attn-SF [16]	104.34	0.043	103.61	0.041
ScaleCrafter [10]	59.40	0.021	57.26	0.018
Ours-TF	**38.99**	**0.015**	**34.73**	**0.013**

Table 2. Quantitative results of different methods on the dataset of *Laion-5B* with $16\times$ image scale adaptation on 2048^2 resolution.

Experiments

Video Generation

Methods	FVD$_r$↓	KVD$_r$↓
Original	688.07	67.17
ScaleCrafter [10]	562.00	44.52
Ours-TF	**553.85**	**33.83**
Full Fine-tuning ($10k$)	721.32	94.57
Full Fine-tuning ($50k$)	531.57	33.61
LORA-R4 ($10k$)	1221.46	263.62
LORA-R32 ($10k$)	959.68	113.07
LORA-R4 ($50k$)	623.72	74.13
LORA-R32 ($50k$)	615.75	76.99
Ours-T ($10k$)	**494.19**	**31.55**

Table 3. Quantitative results of different methods on the dataset of *Webvid-10M* with $4\times$ video scale adaptation on $\boxed{16 \times 512^2}$ resolution (16 frames). $10k$ and $50k$ denote the training steps of each method.

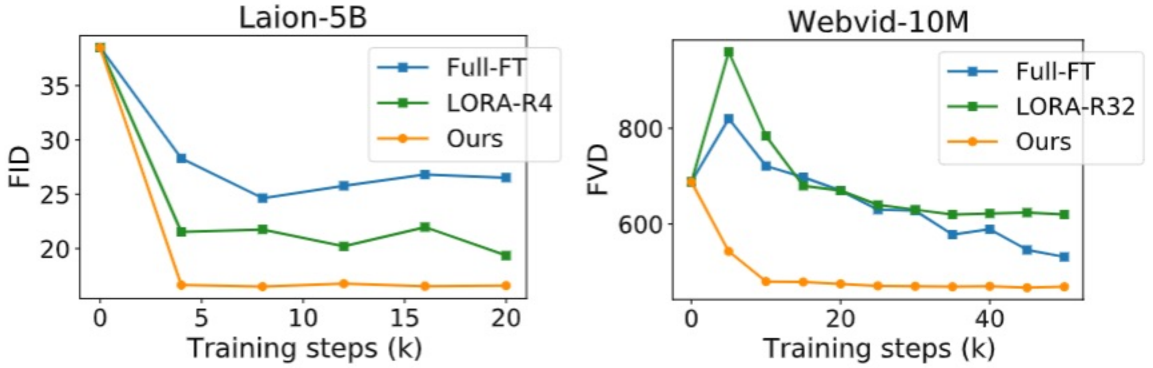


Figure 5. Average FID and FVD scores of three methods every $5k$ iterations on image (Laion-5B) and video (Webvid-10M) datasets. Our observations indicate that our method can rapidly adapt to the higher-resolution domain while maintaining a robust performance among both image and video generation.

Experiments

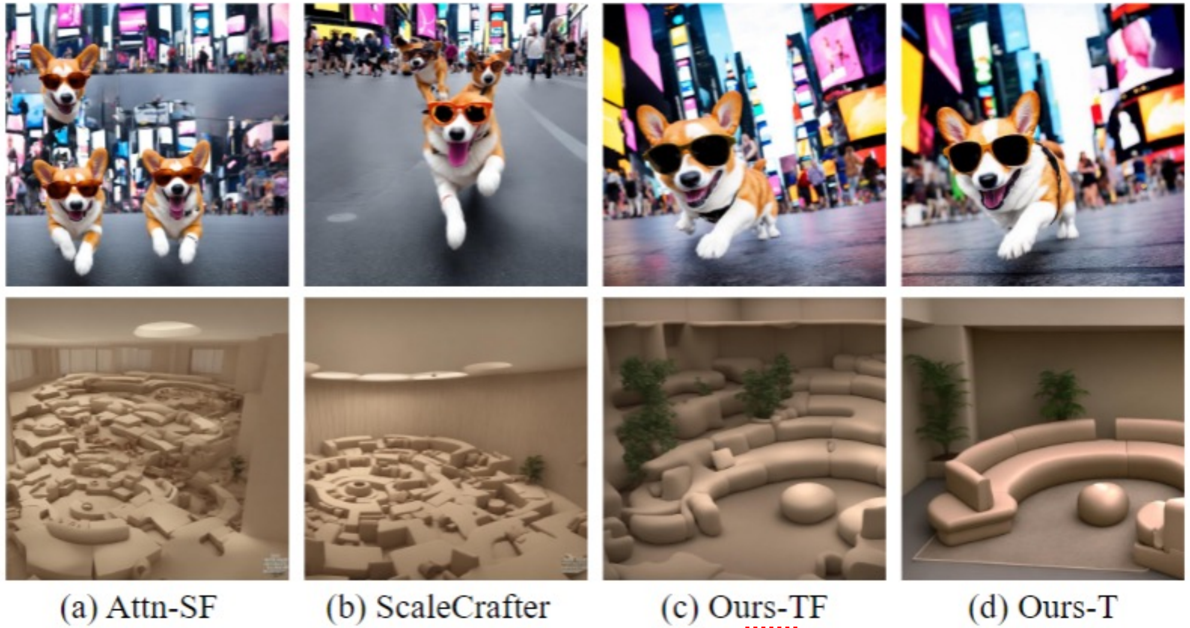


Figure 6. Visual quality comparisons between the training-free methods and ours on higher-resolution adaptation with 1024^2 resolutions. Please zoom in for more details.

Repetition