文生图之SD3

原文链接：https://zhuanlan.zhihu.com/p/686273242

在发布Stable Diffusion 3之后，StabilityAI最近终于放出了SD3的技术报告，相比SD之前的版本，SD3有比较大的改进。首先，SD3是一个基于Rectified Flow的生成模型；其次，SD3引入了T5-XXL来作为text encoder来提升模型的文本理解能力；最后，SD3采用了一个多模态的DiT架构，并且将模型参数量扩展为8B。从目前给出的例子和评测上，SD3在文字渲染和对文本提示词的遵循上，已经达到甚至超过目前STOA的文生图模型如DALL·E 3、Midjourney v6和Ideogram v1。这篇文章将根据SD3的论文分析SD3的具体实现细节。

Prompt: A beautiful painting of flowing colors and styles forming the words "The SD3 research paper is here!", the background is speckled with drops and splashes of paint

改进的RF

SD3相比之前的SD一个最大的变化是采用**Rectified Flow**来作为生成模型，Rectified Flow在Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow被首先提出，但其实也有同期的工作比如Flow Matching for Generative Modeling提出了类似的想法。这里和SD3的论文一样，首先将基于**Flow Matching**来介绍RF，然后再介绍SD3在RF上的具体改进。

Flow Matching

Flow Matching（FM）是建立在continuous normalizing flows的基础上，这里将生成模型定义为一个**常微分方程（ODE）**：

$$dz_t = v(z_t, t)\,dt$$

这里 $t \in [0,1]$ ，而 $v(z_t,t)$ 称之为**向量场（vector field）**。我们用这样的一个ODE来构建一个**概率路径（probability path）** p_t ，它可以实现从一个噪音分布 p_1 到另外一个数据分布 p_0 的转变（可以称之为a flow），注意这里我们在时间上是和FM论文中的定义是相反的，这其实是为了后面和扩散模型统一起来。这里的噪音分布我们采用高斯噪音，即 $p_1 = \mathcal{N}(0,1)$ ，而 p_0 是我们要建模的数据分布 $q(x_0)$ 。一旦我们知道了 $v(z_t,t)$ ，我们就可以用ODE的求解器比如欧拉方法（Euler method）实现从一个噪音到真实数据的生成。这里，我们可以用一个参数为 θ 的神经网络 $v_\theta(z_t,t)$ 来建模向量场，FM的优化目标为：

$$\mathcal{L}_{FM} = \mathbb{E}_{t,p_t(z)}||v_\theta(z,t) - u_t(z)||_2^2$$

这里的 $u_t(z)$ 是目标向量场，它可以产生噪音分布 p_1 到真实数据分布 $q(x_0)$ 的概率路径 $p_t(z)$ 。所以其实FM的优化目标就是直接回归目标向量场。有很多的概率路径可以满足 $p_1 \approx q(x_0)$ ，但是果没有任何先验， $u_t(z)$ 是不可知的，FM的优化目标也就无法实现。

一个解决思路是我们先预先构建一个 $u_t(z)$ ，并让它能够保证我们的目标概率路径 $p_t(z)$ 。为此，FM论文中引入了条件概率路径 $p_t(z|x_0)$ ，这里的条件是真实数据 x_0 ，这个条件概率采用如下的高斯分布：

$$p_t(z|x_0) = \mathcal{N}(z|a_t x_0, b_t^2 I)$$

这个高斯分布的均值为 $a_t x_0$ ，而方差为 b_t ，这里的 a_t 和 b_t 都是和 t 有关的函数，并且是可导的。同时，当 $t=0$ 要满足 $a_0 = 1, b_0 = 0$ ，这样 $p_0(z|x_0) = q(x_0)$ ；而当 $t=1$ 要满足 $a_1 = 0, b_1 = 1$ ，这样 $p_1(z|x_0) = p_1$ 。这样，这里定义的条件概率路径 $p_t(z|x_0)$ 能够保证噪音分布 p_1 到真实数据分布 $q(x_0)$ 的转变。

细心的你可能会发现 $p_t(z|x_0)$ 和扩散模型的扩散过程有相同的形式。其实，引入条件概率路径，就是相当于我们定义了一个前向过程：

$$z_t = a_t x_0 + b_t \epsilon \quad \text{where } \epsilon \sim \mathcal{N}(0, I)$$

后面我们也会看到FM其实是可以看成扩散模型，只是采用了不一样的优化目标（等价于采用不同的loss权重）。

接下来，我们来看一个新的优化目标，那就是Conditional Flow Matching (CFM)目标：

$$\mathcal{L}_{CFM} = \mathbb{E}_{t,q(x_0),p_t(z|x_0)}||v_\theta(z,t) - u_t(z|x_0)||_2^2$$

这里的条件向量场 $u_t(z|x_0)$ 产生条件概率路径 $p_t(z|x_0)$ 。对于CM目标和CFM目标，一个很重要的结论是两者之间只相差一个与参数 θ 无关的常量，这也就意味着：

$$\nabla_\theta \mathcal{L}_{FM}(\theta) = \nabla_\theta \mathcal{L}_{CFM}(\theta)$$

换句话说，使用CFM目标来训练 θ 是和采用CM目标来训练 θ 是等价的。这里我们就不展开证明了，感兴趣的可以看FM论文中的证明。一个直观的解释是，我们采用CFM目标来训练 θ 也是能够达到我们的目标，那就是从噪音分布 p_1 到真实数据分布 $q(x_0)$ ，只不过这里我们人工设定了一个路径 $u_t(z|x_0)$

而已。而且后面我们会看到不同的生成模型的差异除了优化目标之外就在于定义的路径（前向过程）的差异。

虽然 $u_t(z)$ 是不可知的，但是引入条件后的 $u_t(z|x_0)$ 是可以计算出来的：

$$u_t(z|x_0) = z_t' = a_t' x_0 + b_t' \epsilon$$

进一步根据前向过程我们有： $x_0 = (z_t - b_t\epsilon)/a_t$ ，我们将其代入上式，可以得到：

$$u_t(z|x_0) = \frac{a_t'}{a_t} z_t - \epsilon b_t (\frac{a_t'}{a_t} - \frac{b_t'}{b_t})$$

这里我们定义信噪比 $\lambda_t = \log \frac{a_t^2}{b_t^2}$ ，进而有 $\lambda_t' = 2(\frac{a_t'}{a_t} - \frac{b_t'}{b_t})$ ，所以有：

$$u_t(z|x_0) = \frac{a_t'}{a_t} z_t - \frac{b_t}{2} \lambda_t' \epsilon$$

我们将上式代入CFM目标中，就可以得到：

$$\mathcal{L}_{CFM} = \mathbb{E}_{t, q(x_0), p_t(z|x_0), \epsilon \sim \mathcal{N}(0,I)} ||v_\theta(z,t) - \frac{a_t'}{a_t} z + \frac{b_t}{2} \lambda_t' \epsilon||_2^2$$

这里我们对 $v_\theta(z,t)$ 进一步定义为：

$$v_\theta(z,t) = \frac{a_t'}{a_t} z_t - \frac{b_t}{2} \lambda_t' \epsilon_\theta(z,t)$$

代入CFM优化目标可得到：

$$\mathcal{L}_{CFM} = \mathbb{E}_{t, q(x_0), p_t(z|x_0), \epsilon \sim \mathcal{N}(0,I)} \left(-\frac{b_t}{2} \lambda_t'\right)^2 ||\epsilon_\theta(z,t) - \epsilon||_2^2$$

此时相当于神经网络变成了预测噪音，这和扩散模型DDPM预测噪音是一样的，但是优化目标的多了一个和 t 有关的权重系数。所以，FM其实可以看成一个采用不同的权重系数的扩散模型。

Google的工作Understanding Diffusion Objectives as the ELBO with Simple Data Augmentation提出了一个统一的视角，即不同的生成模型包括DDPM，SDE，EDM以及FM等的优化目标都可以统一为：

$$\mathcal{L}_w(x_0) = -\frac{1}{2} \mathbb{E}_{t \sim \mathcal{U}(0,1), \epsilon \sim \mathcal{N}(0,I)} \left[w_t \lambda_t' ||\epsilon_\theta(z_t, t) - \epsilon||^2 \right]$$

不同的生成模型所采用的优化目标不同，等价于采用不同的权重 w_t 。对于DDPM所采用的 $\mathcal{L}_{simple}$ ，这里 $w_t = -2/\lambda_t'$ 。而对于FM的 $\mathcal{L}_{CFM}$ ，有 $w_t = -\frac{1}{2} \lambda_t' b_t^2$ 。

更具体地说，不同类型的生成模型差异在于前向过程和预测目标的差异。不同的前向过程采用不同 a_t 和 b_t ，导致不同的概率路径。而预测目标可以为预测噪音 ϵ （DDPM），预测分数 s （SDE），以及预测向量场 v （FM）等等。但是它们都可以最终统一为基于预测噪音 ϵ 的优化目标，只是权重 w_t 的差异。

Rectified Flow

在FM中，作者给出了一个基于最优传输（ Optimal Transport）具体的前向过程：

$$z_t = (1-t)x_0 + ((1-t)\sigma_{min} + t)\epsilon$$

当 $\sigma_{min} = 0$ ，我们就可以得到和Rectified Flow中一样的前向过程：

$$z_t = (1-t)x_0 + t\epsilon$$

RF的前向过程一个特点是 z_t 由数据 x_0 和噪音 ϵ 线性插值得到，这也意味我们人工定义的概率路径是一条直线。直线的一个好处是采样时我们可以步子迈大一点，这就相当于我们可以减少采样的总步数。关于理论的分析涉及到最优传输，感兴趣的话可以看看论文。

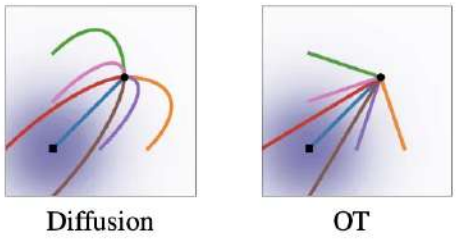


Figure 3: Diffusion and OT trajectories.

对于RF，有 $z_t' = -x_0 + \epsilon$ ，所以其优化目标就变成了：

$$\mathcal{L}_{RF} = \mathbb{E}_{t,q(x_0),p_t(z|x_0),\epsilon\sim\mathcal{N}(0,I)}||v_\theta(z,t) - (\epsilon - x_0)||_2^2$$

可以看到，最终RF的损失函数是非常简单的。如果将RF转成 $\mathcal{L}_w(x_0)$ ，其对应的 $w_t = -\frac{1}{2}\lambda_t' b_t^2 = \frac{t}{1-t}$ 。

SD3论文中除了实验RF模型外，还对其它模型做了对比实验，这里也需要简单介绍一下。

首先是之前版本的SD所采用的**(LDM-)Linear**，LDM是基于DDPM，但和DDPM采用了不同的noise schedule。DDPM是基于离散时间 $t = 0, \ldots, T-1$ 的扩散模型，给定扩散系数 β_0 和 β_T ，$\beta_t = \beta_0 + \frac{t}{T-1}(\beta_{T-1} - \beta_0)$ （DDPM的noise schedule是线性的）。对于LDM，$\beta_t = \left(\sqrt{\beta_0} + \frac{t}{T-1}(\sqrt{\beta_{T-1}} - \sqrt{\beta_0})\right)^2$ 。根据 β_t ，可以得到：

$$a_t = (\prod_{s=0}^{t}(1-\beta_s))^{\frac{1}{2}}, b_t = \sqrt{1-a_t^2}$$

除了线性noise schedule，I-DDPM还提出了cosine noise schedule，其前向过程可以定义为（采用连续时间）：

$$z_t = \cos(\frac{\pi}{2}t)x_0 + \sin(\frac{\pi}{2}t)\epsilon$$

除了此外，SD3还实验了EDM，但这里我们不再展开了。

改进的采样方法

这里所说的采样是指的训练过程对时间步 t 的采样，由于 t 是和信噪比SNR正相关的，所以也可以说是对SNR的采样。对于RF，其默认使用均匀分布 $t \sim \mathcal{U}(0,1)$ 进行采样，这也就是说各个时间步 t 是同等对待的。但是SD3论文中认为不同时间步的任务难度是一样：两边相对容易，而中间是比较难的。所以，这里是设计了一些新的采样方法来提高中间时间步的权重。改变采样的分布，等价于改变权重系数：

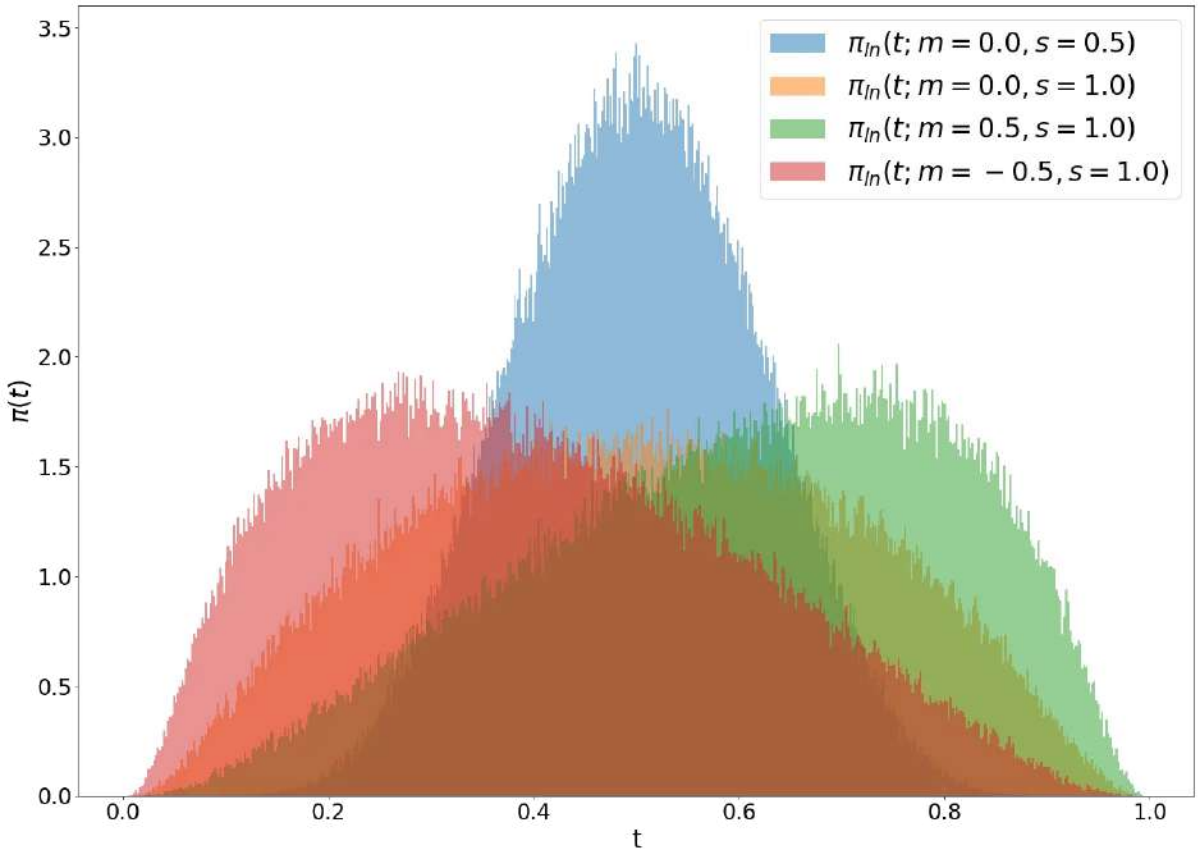
$$w_t^\pi = \frac{t}{1-t}\pi(t)$$

这里的 $\pi(t)$ 是采样 t 所遵循的概率分布，当使用均匀分布 $t \sim \mathcal{U}(0,1)$ 时， $\pi(t) = 1$ 。下面我们介绍一下SD3论文中所实验的几种采样方法。

第一个采样方法是**Logit-Normal Sampling**，这是采用Logit-Normal分布，所谓的Logit-Normal分布是指变量的logit满足正态分布，对于Logit-Normal分布，其概率密度为：

$$\pi_{\ln}(t;m,s) = \frac{1}{s\sqrt{2\pi}}\frac{1}{t(1-t)}\exp\Big(-\frac{(\text{logit}(t)-m)^2}{2s^2}\Big)$$

这里 $\text{logit}(t) = \log\dfrac{t}{1-t}$ 。其中参数 m 可以控制 t 的偏向（其中 $m=0$ 时， $t=0.5$ 是分布的峰值），参数 s 控制分布的宽度（或者说是胖瘦）。下面是不同的参数下分布的可视化：

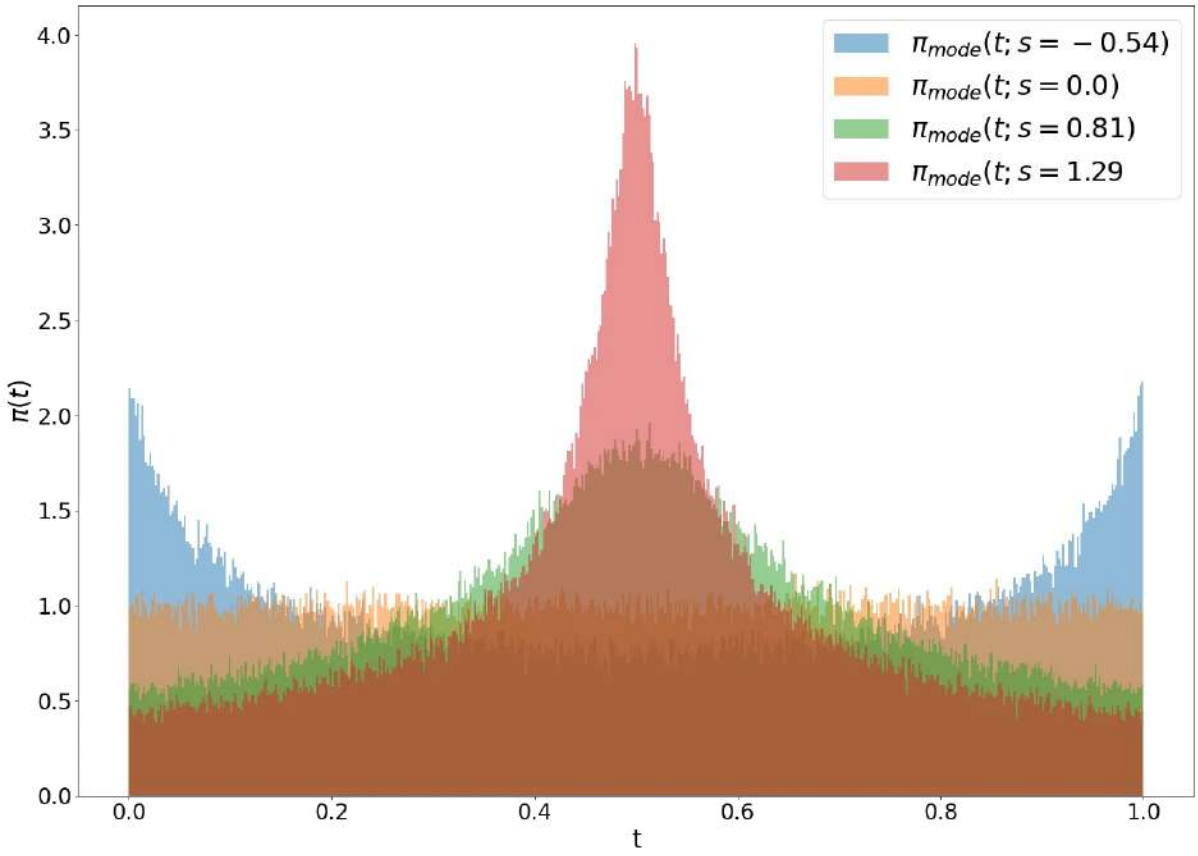


在采样过程中，我们可以先基于正态分布 $u \sim \mathcal{N}(u; m, s)$ 采样出一个 u ，然后再转成 $t = \frac{e^u}{1+e^u}$ 。

第二个采样方法是**Mode Sampling with Heavy Tails**。Logit-Normal分布的一个问题是两边 $t = 0$ 和 $t = 1$ 附近基本采样不到，这个可能会对性能有一定的影响。所以这个第二个采样方法是基于一个重尾分布。首先我们用定义如下的函数：

$$f_{\text{mode}}(u; s) = 1 - u - s \cdot \left(\cos^2\left(\frac{\pi}{2}u\right) - 1 + u\right)$$

这里 $-1 \leq s \leq \frac{2}{\pi - 2}$ ，此时函数是单调的，我们可以通过 $u \sim [0, 1], t = f_{\text{mode}}(u; s)$ 来采样时间步 t 。根据变量变换定理，有 $\pi_{\text{mode}}(t; s) = \pi(u)\left|\frac{d}{dt} f^{-1}_{\text{mode}}(t)\right| = \left|\frac{d}{dt} f^{-1}_{\text{mode}}(t)\right|$ 。这里的参数 s 控制分布是偏向中间（>0）还是偏向两边（<0），当 $s = 0$ 时，此时就相当于均匀分布了，即 $\pi_{\text{mode}}(t; 0) = 1$ 。下面是不同 s 下的分布可视化。



最后一个采样方法是**CosMap**。这里其实是想实现下RF下的cosine schedule ， 我们可以求解一个映射 $f : u \mapsto f(u) = t,\ u \in [0, 1]$ ，让SNR和cosine schedule是一样的，即：

$$2 \log \frac{\cos(\frac{\pi}{2}u)}{\sin(\frac{\pi}{2}u)} = 2 \log \frac{1 - f(u)}{f(u)}$$

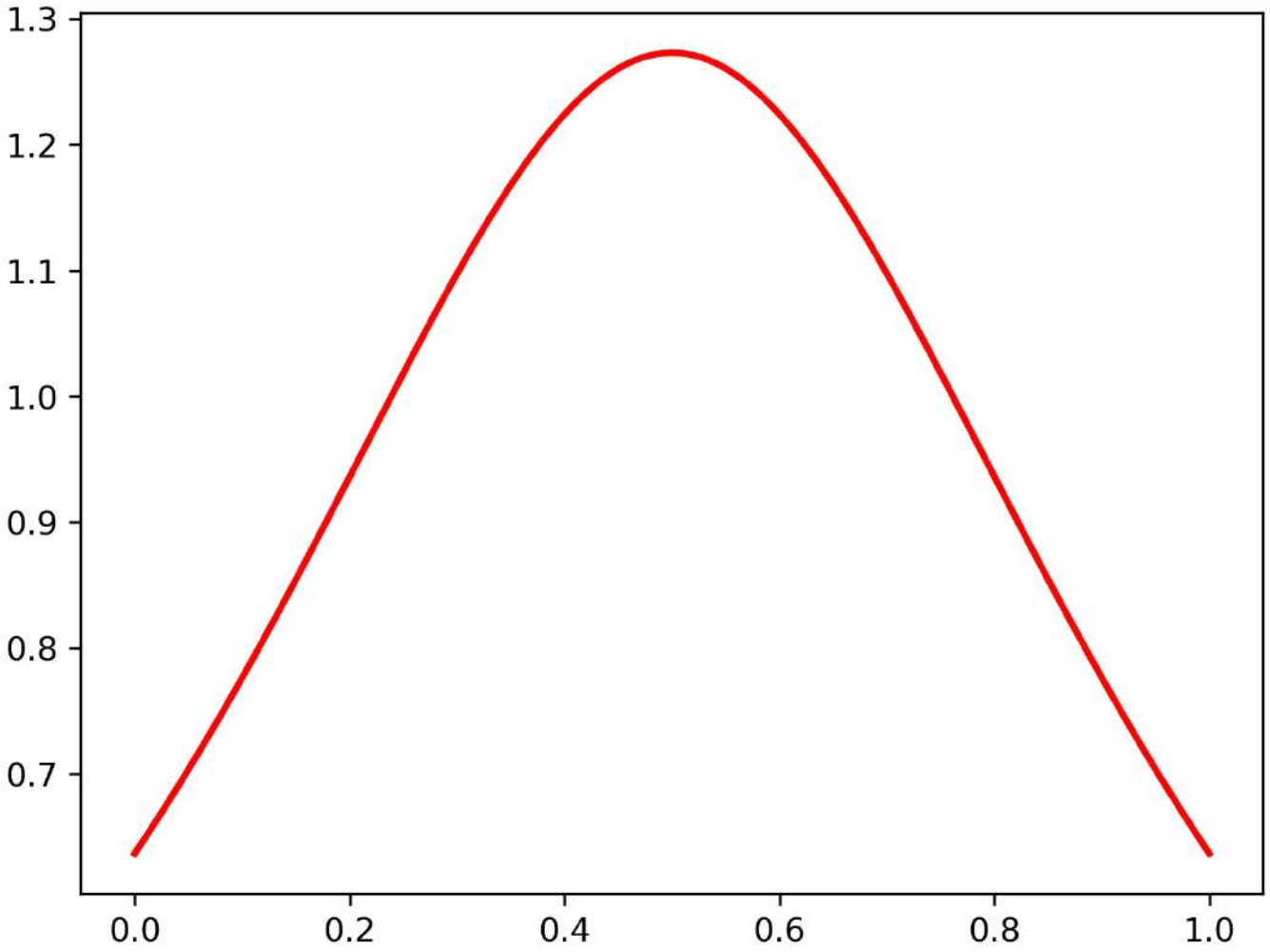
通过上述等式可得：

$$t = f(u) = 1 - \frac{1}{\tan(\frac{\pi}{2}u) + 1}$$

同样根据变量变换定理，我们可以得到 t 的概率密度：

$$\pi_{\text{CosMap}}(t) = \left| \frac{d}{dt} f^{-1}(t) \right| = \frac{2}{\pi - 2\pi t + 2\pi t^2}$$

这里我们可以画出这个分布，如下所示，它也是中间概率密度高：



对比实验

为了验证RF是否在文生图上是有效的，SD3论文中做了一系列的对比实验，实验的模型共包括61个，分别是：

- 采用 ϵ 和 v 优化目标，同时noise schedule采用linear和cosine，这共4个配置：eps/linear，v/linear，eps/cos, v/cos，其中eps/linear就是LDM所采用的配置。
- 采用RF和 $\pi_{\mathrm{mode}}(t;s)$ ，这里记为rf/mode(s)，其中其中 s 在−1∼1.75之间均匀选取7个值，另外还包含一个 $s=0$ 的配置，这其实就是原来的rf。所以这组总共8个配置。
- 采用RF和 $\pi_{\ln}(t;m,s)$ ，这里记为rf/lognorm(m, s)，其中在 $m\sim[-1,1]$ 和 $s\sim[0.2,2.2]$ 以网格方式选择30组 (m,s) 。
- 采用RF和 $\pi_{\mathrm{CosMap}}(t)$ ，这里记为rf/cosmap。
- 采用EDM，记为edm(添加 TeX 公式)，这两个参数决定EDM的SNR，其中在 $P_m\sim[-1.2,1.2]$ 和 $P_s\sim[0.6,1.8]$ 均匀选择15组。

- 采用EDM，但是schedule分别设置为和rf以及v/cos的SNR加权匹配，这两个配置分别记为edm/rf和edm/cos。

每个模型的实验配置如下：

- **训练数据集**：ImageNet和CC12M两个数据集，其中ImageNet数据通过"a photo of a <class name>"构造成文本–图像对数据集。
- **评测指标：**CLIP score和FID（这里的FID采用CLIP来计算特征，而不是基于Inception V3），同时还基于validation loss选择模型。
- **评测数据集**：COCO–2014验证集。
- **采样器设置**：推理阶段均采用欧拉方法，共包括不同steps和CFG scale的6个配置，50 steps（CFG scale为1.0, 2.5, 5.0）以及CFG scale为5.0的5, 10, 25 steps。
- **权重**：非EMA和EMA权重。

每个实验用EMA权重在不同的训练steps基于validation loss最小来确定最优的模型。这里2个训练数据集+6个采样器设置+2套参数共产生24个组合，所以每个模型也会得到24个评测结果。由于评测指标是2个，所以采用多目标优化中非支配排序算法（基于Pareto最优）来进行排序。每一种配置（24种）单独进行排序，然后取平均值。下表展示了不同模型的rank结果（这里只展示每组配置的top 2）：

	rank averaged over		
variant	all	5 steps	50 steps
rf/lognorm(0.00, 1.00)	1.54	1.25	1.50
rf/lognorm(1.00, 0.60)	2.08	3.50	2.00
rf/lognorm(0.50, 0.60)	2.71	8.50	1.00
rf/mode(1.29)	2.75	3.25	3.00
rf/lognorm(0.50, 1.00)	2.83	1.50	2.50
eps/linear	2.88	4.25	2.75
rf/mode(1.75)	3.33	2.75	2.75
rf/cosmap	4.13	3.75	4.00
edm(0.00, 0.60)	5.63	13.25	3.25
rf	5.67	6.50	5.75
v/linear	6.83	5.75	7.75
edm(0.60, 1.20)	9.00	13.00	9.00
v/cos	9.17	12.25	8.75
edm/cos	11.04	14.25	11.25
edm/rf	13.04	15.25	13.25
edm(-1.20, 1.20)	15.58	20.25	15.00

Table 1. **Global ranking of variants.** For this ranking, we apply non-dominated sorting averaged over EMA and non-EMA weights, two datasets and different sampling settings.

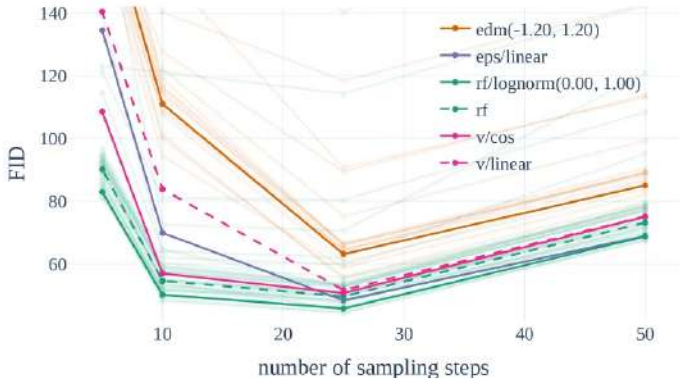
可以看到**rf/lognorm(0.00, 1.00)**是综合rank最高的，而且在5 steps和50 steps下也可以取得较好的rank。这里所采用的lognorm(0.00, 1.00)的时间采样方法也恰好是偏向中间时间步的，这说明对中间时间步加权是重要且有效的。这里也可以看到未改进的rf效果上反而是不如LDM所采用的eps/linear，而且经典的eps/linear的rank也仅次于几个改进的rf。

下表展示了不同的模型在25 steps下具体的CLIP score和FID，rf/lognorm(0.00, 1.00)两个数据集均表现不错，而经典的eps/linear其实也不差。

	ImageNet		CC12M	
variant	CLIP	FID	CLIP	FID
rf	0.247	49.70	0.217	94.90
edm(-1.20, 1.20)	0.236	63.12	0.200	116.60
eps/linear	0.245	48.42	0.222	*90.34*
v/cos	0.244	50.74	0.209	97.87
v/linear	0.246	51.68	0.217	100.76
rf/lognorm(0.50, 0.60)	**0.256**	80.41	0.233	120.84
rf/mode(1.75)	*0.253*	**44.39**	0.218	94.06
rf/lognorm(1.00, 0.60)	0.254	114.26	**0.234**	147.69
rf/lognorm(-0.50, 1.00)	0.248	45.64	0.219	**89.70**
rf/lognorm(0.00, 1.00)	0.250	*45.78*	*0.224*	89.91

Table 2. **Metrics for different variants.** FID and CLIP scores of different variants with 25 sampling steps. We highlight the **best**, second best, and *third best* entries.

我们可以进一步去观察不同steps下各个模型的表现，如下图所示：

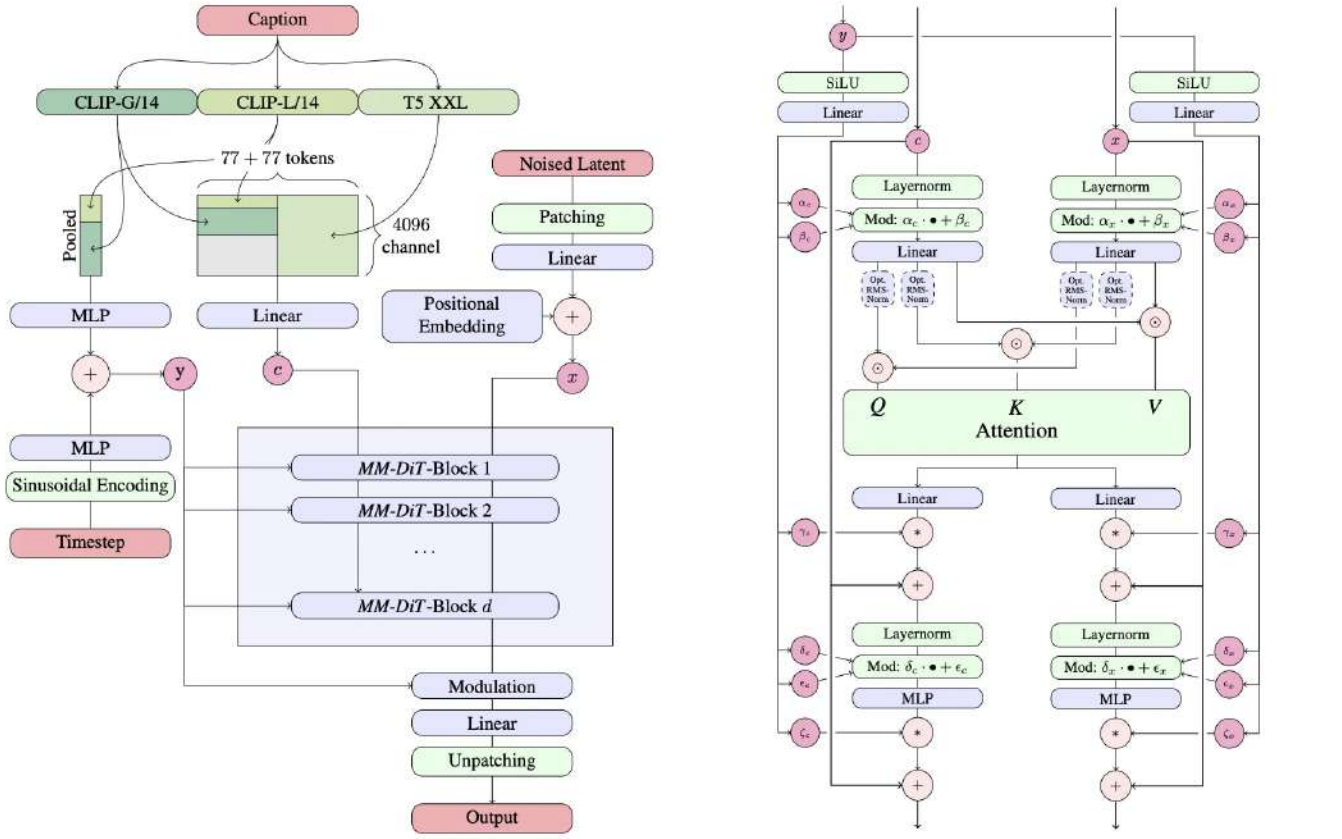


可以看到rf模型在steps比较小时展现比较明显的优势，说明rf模型可以减少推理阶段的采样步数。当steps增加时，rf不如eps/linear，但是改进后的rf/lognorm(0.00, 1.00)依然能够超过eps/linear。

总结：RF模型推理高效，但是通过改进时间采样方法对中间时间步加权能进一步提升效果，这里基于lognorm(0.00, 1.00)的采样方法从实验看是最优的。

多模态DiT

SD3除了采用改进的RF，另外一个重要的改进就是采用了一个多模态DiT。多模态DiT的一个核心对图像的latent tokens和文本tokens拼接在一起，并采用两套独立的权重处理，但是在attention时统一处理。整个架构图如下所示：



(a) Overview of all components. (b) One *MM-DiT* block

Figure 2. **Our model architecture.** Concatenation is indicated by $\odot$ and element-wise multiplication by $*$. The RMS-Norm for Q and K can be added to stabilize training runs. Best viewed zoomed in.

改进的autoencoder

这里的MM-DiT和DiT一样，依然是使用一个autoencoder（VAE）来将图像编码为latent，然后将latent转成patches，送入transformer处理。之前版本的SD所使用的autoencoder是将一个 $H \times W \times 3$ 的图像编码为 $\frac{H}{8} \times \frac{W}{8} \times d$ 的latent，这里的 $d = 4$ ，这个压缩还是比较狠的，带来的不利影响是容易产生小物体畸变（比如人眼，文子等）。所以SD3通过增加 d 来提升autoencoder的重建质量。下面是不同的 d 的定量评估：

Metric	4 chn	8 chn	16 chn
FID (↓)	2.41	1.56	**1.06**
Perceptual Similarity (↓)	0.85	0.68	**0.45**
SSIM (↑)	0.75	0.79	**0.86**
PSNR (↑)	25.12	26.40	**28.62**

Table 3. **Improved Autoencoders.** Reconstruction performance metrics for different channel configurations. The downsampling factor for all models is $f = 8$.

当 $d = 16$ 时，autoencoder的性能相比的 $d = 4$ 有一个比较大的提升，所以SD3使用16通道的autoencoder。要注意，虽然增加通道并不会对生成模型（UNet或者DiT）的参数带来大的影响（只需要修改网络第一层和最后一层的通道数），但是会增加任务的难度，当通道数从4增加到16，网络要拟合的内容增加了4倍，这也意味模型需要增加参数来提供足够的容量。SD3论文中的一个实验对比结果如下所示：

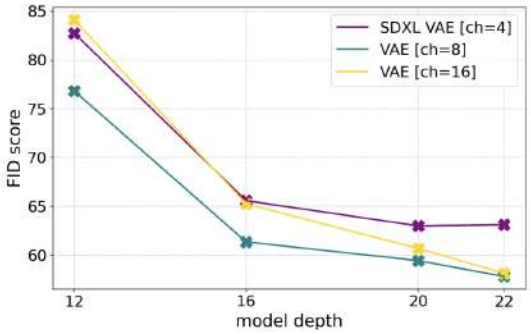


Figure 10. FID scores after training flow models with different sizes (parameterized via their depth) on the latent space of different autoencoders (4 latent channels, 8 channels and 16 channels) as discussed in Section 5.2.1. As expected, the flow model trained on the 16-channel autoencoder space needs more model capacity to achieve similar performance. At depth $d = 22$, the gap between 8-chn and 16-chn becomes negligible. We opt for the 16-chn model as we ultimately aim to scale to much larger model sizes.

当模型参数小时，16通道的autoencoder并没有比4通道的autoencoder更好，但当模型参数增加时，16通道的autoencoder的优势慢慢展示出来，当模型深度到22时，16通道的autoencoder明显优于4通道的autoencoder。不过这里8通道的autoencoder在FID上也不差于16通道的autoencoder，但FID只是图像质量的一个间接评价指标，并不能提现图像细节的差异，从重建效果上看，16通道的autoencoder应该优势更明显，而且当模型变大后，上限更高。

比较类似的是，之前Meta的文生图模型Emu也采用16通道的autoencoder来提升图像细节。

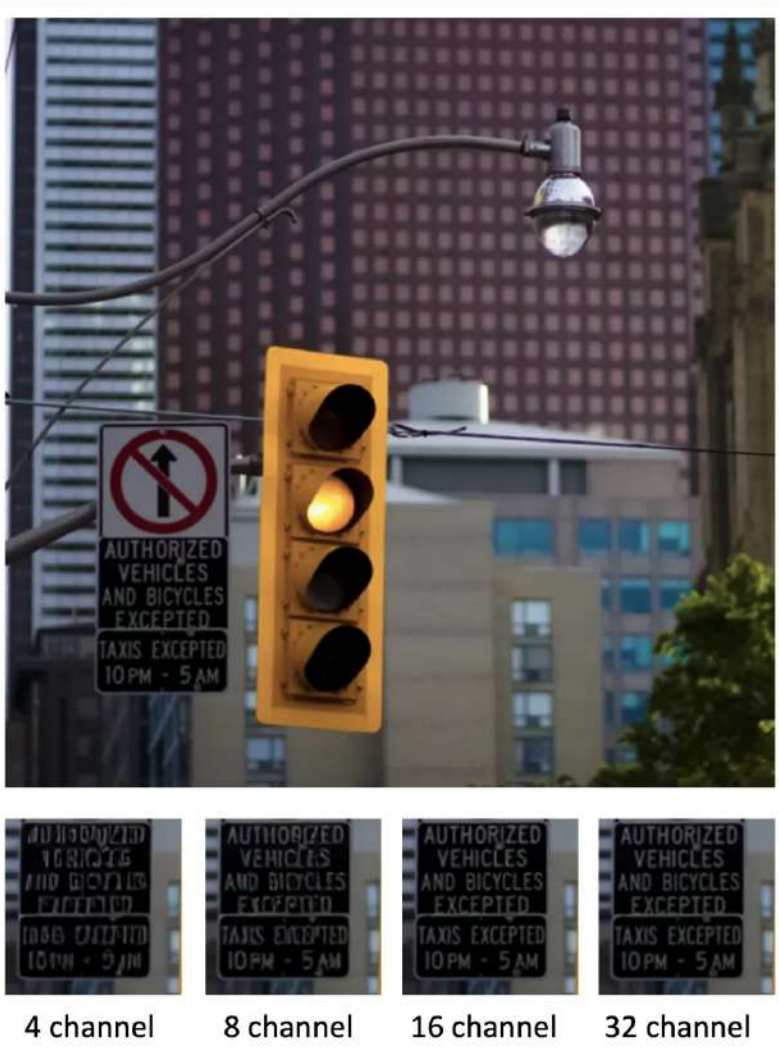


Figure 3. **Autoencoder.** The visual quality of the reconstructed images for autoencoders with different channel sizes. While keeping all other architecture layers the same, we only change the latent channel size. We show that the original 4-channel autoencoder design [27] is unable to reconstruct fine details. Increasing channel size leads to much better reconstructions. We choose to use a 16-channel autoencoder in our latent diffusion model.

而DALLE-3则是通过训练一个基于扩散模型的latent decoder来解决4通道autoencoder的问题，但是不如直接采用16通道的autoencoder，直接从源头解决问题。

文本编码器

SD3的text encoder包含3个预训练好的模型：

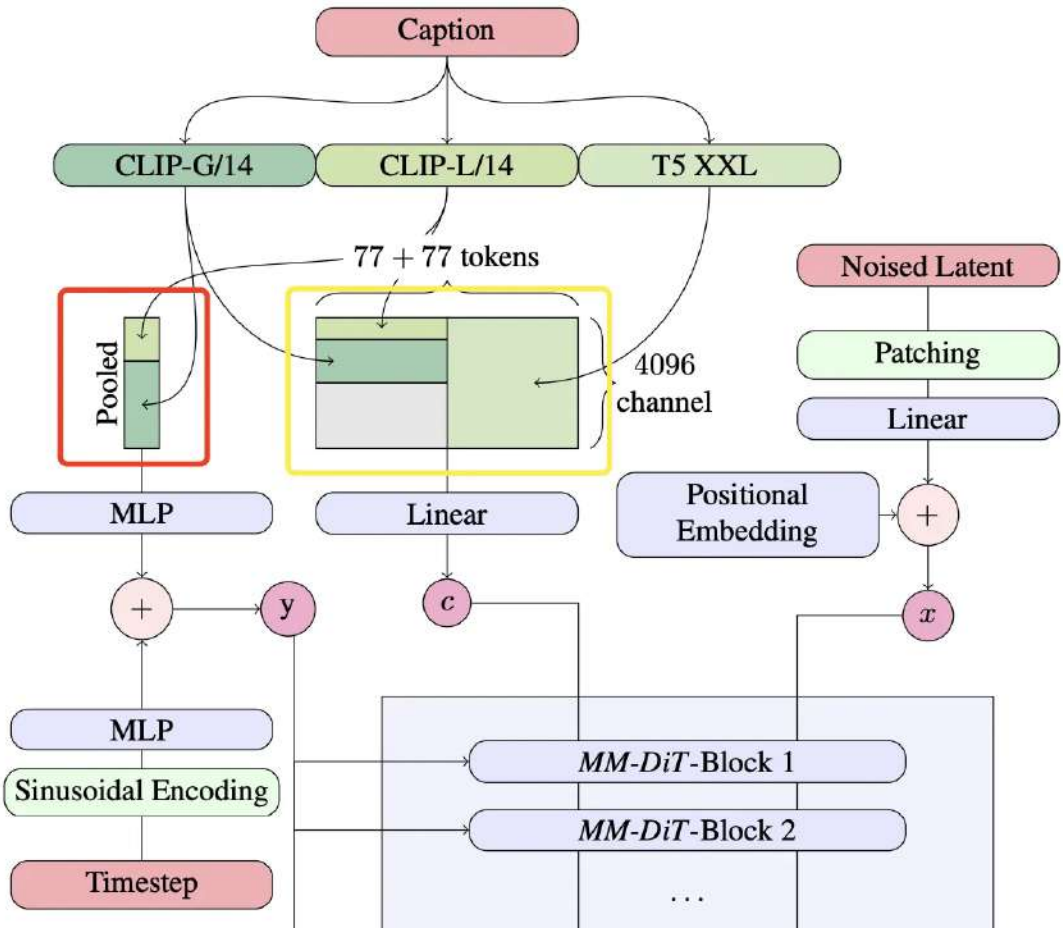
- CLIP ViT-L：参数量约124M
- OpenCLIP ViT-bigG：参数量约695M
- T5-XXL encoder：参数量约4.7B

SD 1.x模型的text encoder使用CLIP ViT-L，SD 2.x模型的text encoder采用OpenCLIP ViT-H，而SDXL的text encoder使用CLIP ViT-L + OpenCLIP ViT-bigG。这次SD3更上一个台阶，加上了一个更大的T5-XXL encoder。谷歌的Imagen最早使用T5-XXL encoder作为文生图模型的text encoder，并证明预训练好的纯文本模型可以实现更好的文本理解能力，后面的工作，如NVIDIA的eDiff-I和Meta的Emu采用T5-XXL encoder + CLIP作为text encoder，OpenAI的DALL-E 3也采用T5-XXL encoder。SD3加入T5-XXL encoder也是模型在文本理解能力特别是文字渲染上提升的一个关键。

具体地，SD3总共提取两个层面的特征。

首先提取两个CLIP text encoder的pooled embedding，它们是文本的全局语义特征，维度大小分别是768和1024，两个embedding拼接在一起得到2048的embedding，然后经过一个MLP网络之后和timestep embedding相加。

然后是文本细粒度特征。这里也先分别提取两个CLIP模型的倒数第二层的特征，拼接在一起可以得到77x2048维度的CLIP text embeddings；同样地也从T5-XXL encoder提取最后一层的特征T5 text embeddings，维度大小是77x4096（这里也限制token长度为77）。然后对CLIP text embeddings使用zero-padding得到和T5 text embeddings同维度的特征。最后，将padding后的CLIP text embeddings和T5 text embeddings在token维度上拼接在一起，得到154x4096大小的混合text embeddings。text embeddings将通过一个linear层映射到与图像latent的patch embeddings同维度大小，并和patch embeddings拼接在一起送入MM-DiT中。

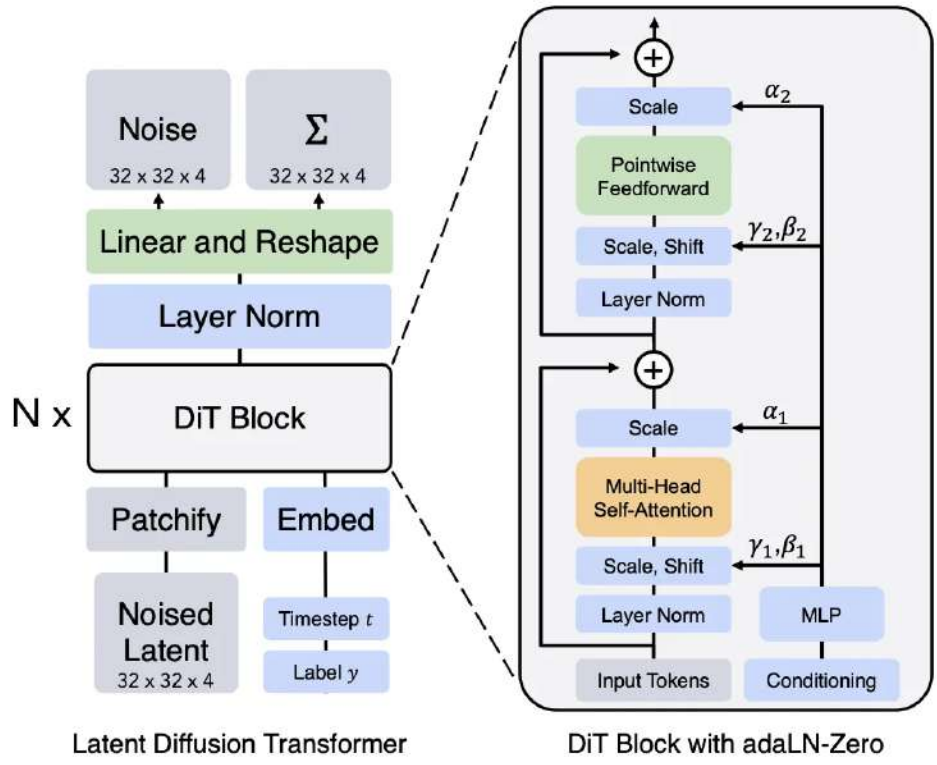


采用CLIP+T5-XXL encoder相比单独的T5-XXL encoder可能带来性能增益，但是一个不利的影响是CLIP text encoder只能默认编码77 tokens长度的文本，这也限制了T5-XXL encoder的token长度（T5-XXL encoder能够编码512 tokens）。DALL-E 3可以输入比较长的文本，而这里的SD3默认只能处理77 tokens长度的文本。

MM-DiT

MM-DiT和DiT一样也是处理图像latent空间，这里先对图像的latent转成patches，这里的patch size=2x2，和DiT的默认配置是一样的。patch embedding再加上positional embedding送入transformer中。

这里的重点是如何处理前面说的文本特征。对于CLIP pooled embedding可以直接和timestep embedding加在一起，并像DiT中所设计的adaLN-Zero一样将特征插入transformer block。



具体的实现代码如下所示：

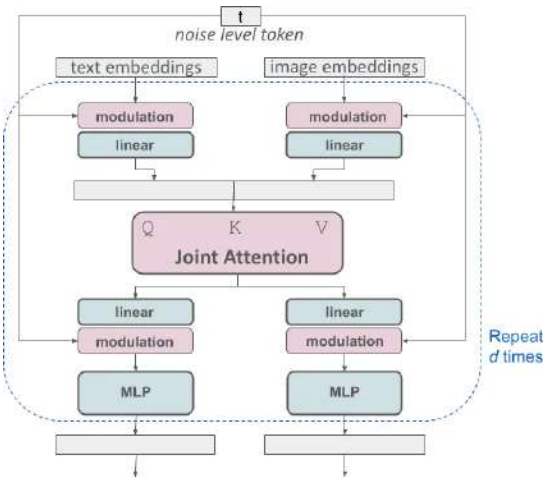
```python
def modulate(x, shift, scale):
    return x * (1 + scale.unsqueeze(1)) + shift.unsqueeze(1)

class DiTBlock(nn.Module):
    """
    A DiT block with adaptive layer norm zero (adaLN-Zero) conditioning.
    """
    def __init__(self, hidden_size, num_heads, mlp_ratio=4.0, **block_kwargs):
        super().__init__()
        self.norm1 = nn.LayerNorm(hidden_size, elementwise_affine=False, eps=1e-6)
        self.attn = Attention(hidden_size, num_heads=num_heads, qkv_bias=True, **block_kwargs)
        self.norm2 = nn.LayerNorm(hidden_size, elementwise_affine=False, eps=1e-6)
        mlp_hidden_dim = int(hidden_size * mlp_ratio)
        approx_gelu = lambda: nn.GELU(approximate="tanh")
        self.mlp = Mlp(in_features=hidden_size, hidden_features=mlp_hidden_dim, act_layer=approx_gelu, drop=0)
        self.adaLN_modulation = nn.Sequential(
            nn.SiLU(),
            nn.Linear(hidden_size, 6 * hidden_size, bias=True)
        )

    def forward(self, x, c):
        shift_msa, scale_msa, gate_msa, shift_mlp, scale_mlp, gate_mlp = self.adaLN_modulation(c).chunk(6, dim=1)
        x = x + gate_msa.unsqueeze(1) * self.attn(modulate(self.norm1(x), shift_msa, scale_msa))
        x = x + gate_mlp.unsqueeze(1) * self.mlp(modulate(self.norm2(x), shift_mlp, scale_mlp))
        return x

```

对于序列的text embeddings，常规的处理方式是增加cross attention层来处理，其中text embeddings作为attention的keys和values，比如SD的UNet以及PIXART-α（基于DiT）。但是SD3是直接将text embeddings和patch embeddings拼在一起处理，这样不需要额外引入cross-attention。由于text和image属于两个不同的模态，这里采用两套独立的参数来处理，即所有transformer层的学习参数是不共享的，但是共用一个self-attention来实现特征的交互。这等价于采用两个transformer模型来处理文本和图像，但在attention层连接，所以这是一个多模态模型，称之为MM-DiT。



MM-DiT和之前文生图模型的一个区别是文本特征不再只是作为一个条件，而是和图像特征同等对待处理。论文中也基于CC12M数据集将MM-DiT和其它架构做了对比实验，这里对比的模型有DiT（这里的DiT是指的不引入cross-attention，直接将text tokens和patches拼接，但只有一套参数），CrossDiT（额外引入cross-attention），UViT（UNet和transformer混合架构），还有3套参数的MM-DiT（CLIP text tokens，T5-XXL text tokens和patches各一套参数）。不同架构的模型表现如下所示：

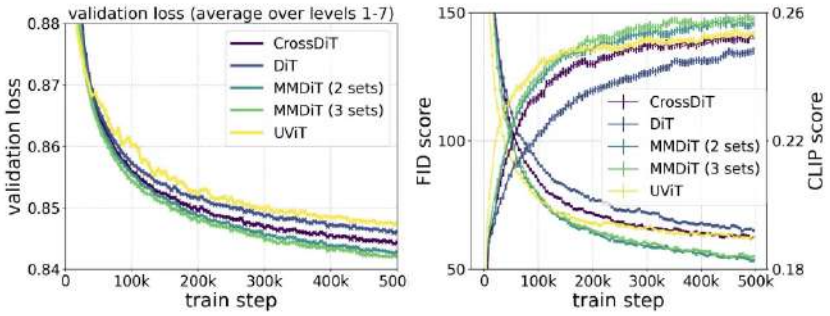


Figure 4. **Training dynamics of model architectures.** Comparative analysis of *DiT*, *CrossDiT*, *UViT*, and *MM-DiT* on CC12M, focusing on validation loss, CLIP score, and FID. Our proposed *MM-DiT* performs favorably across all metrics.

可以看到MM-DiT是优于其它架构的，其中3套参数的MM-DiT略好于2套参数的MM-DiT，最终还是选择参数量更少的2套参数的MM-DiT。不过，这里和其它架构的对比是否保证了同参数大小，否则实验就显得有点不公平了。

MM-DiT的模型参数主要是模型的深度 d ，即transformer block的数量，此时对应的模型中间特征的维度大小是 $64 \cdot d$ 。这意味着当模型的深度 d 增大为 $r \cdot d$ ，模型的参数量会增大 r^3 。比如深度为24的MM-DiT参数量为2B，最大的MM-DiT深度为38，其参数量为 $2B * (38/24)^3 \approx 8B$ 。

QK-Normalization

为了提升混合精度训练的稳定性，MM-DiT的self-attention层还采用了QK-Normalization。当模型变大，而且在高分辨率图像上训练时，attention层的attention-logit（Q和K的矩阵乘）会变得不稳定，导

致训练出现NAN。这里的解决方案是采用RMSNorm（简化版LayerNorm）对attention的Q和K进行归一化。

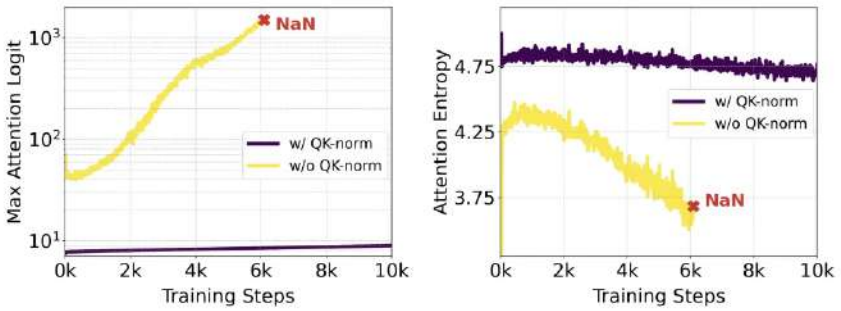
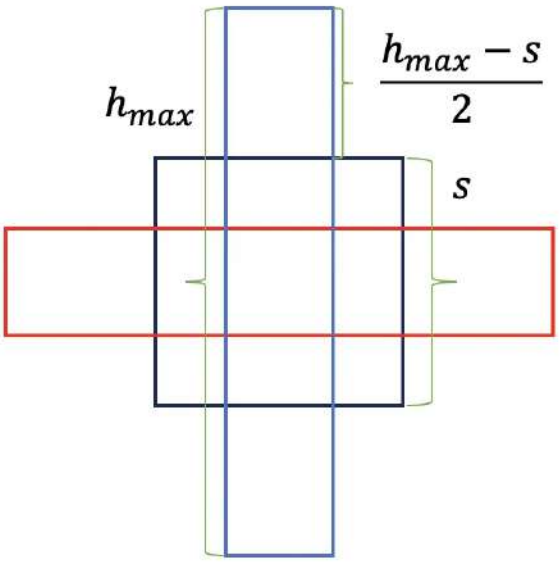


Figure 5. **Effects of QK-normalization.** Normalizing the Q- and K-embeddings before calculating the attention matrix prevents the attention-logit growth instability (*left*), which causes the attention entropy to collapse (*right*) and has been previously reported in the discriminative ViT literature (Dehghani et al., 2023; Wortsman et al., 2023). In contrast with these previous works, we observe this instability in the last transformer blocks of our networks. Maximum attention logits and attention entropies are shown averaged over the last 5 blocks of a 2B (d=24) model.

变尺度位置编码

MM-DiT的位置编码和ViT一样采用2d的frequency embeddings（两个1d frequency embeddings进行concat）。SD3先在256x256尺寸下预训练，但最终会在以1024x1024为中心的多尺度上微调，这就需要MM-DiT的位置编码需要支持变尺度。SD3采用的解决方案是插值+扩展。

这里假定我们的目标分辨率的像素量为 S^2 ，各个尺寸的图像满足 $H \times W \approx S^2$（比如1024x1024，512x2048，2048x512），其中图像的宽和高最大分别为 $H_{\max}$ 和 $W_{\max}$ 。如果换算为MM-DiT的patches，有 $h_{\max} = H_{\max}/16, w_{\max} = W_{\max}/16, s = S/16$ ，因为autoencoder下采样8x，而patch size为2x2，所以最终下采样16x。预训练模型的位置编码是在256x256下训练的，我们可以先通过插值的方式将位置编码应用到 $S \times S$ 尺度上，此时相当于位置 p 处的网格值为 $p \cdot \frac{256}{S}$ ，进一步地，我们可以将其扩展支持最大的宽和高，以高为例子，这里有 $(p - \frac{h_{\max} - s}{2}) \cdot \frac{256}{S}$ 。对于不同的尺寸，我们只需要center crop出对应的2d网格进行embedding得到位置编码。下面的一个比较直观的示意图：



timestep schedule的shift

对高分辨率的图像，如果采用和低分辨率图像的一样的noise schedule，会出现对图像的破坏不够的情况，如下图所示（图源自On the Importance of Noise Scheduling for Diffusion Models）：

(a) 64×64 (b) 128×128 (c) 256×256 (d) 512×512 (e) 1024×1024

Figure 2: Noised images ($\boldsymbol{x}_t = \sqrt{\gamma}\boldsymbol{x}_0 + \sqrt{1-\gamma}\boldsymbol{\epsilon}$) with the same noise level ($\gamma = 0.7$). We see that higher resolution natural images tend to exhibit higher degree of redundancy in (nearby) pixels, therefore less information is destroyed with the same level of independent noise.

一个解决办法是对noise schedule进行偏移，对于RF模型来说，就是timestep schedule的shift。下面我们来理论分析如何进行shift。假定要处理的图像包含 $n = H \times W$ 个像素，但它是一个常量图像，所有的像素值均为 c 。根据RF的前向过程，我们有 $z_t = (1-t)c\mathbf{1} + t\epsilon$ ，这里 $\mathbf{1}, \epsilon \in \mathbb{R}^n$ 。 z_t 可以产生 n 个观察变量 $Y = (1-t)c + t\eta$ ，我们可以计算出均值和标准差：

$\mathbb{E}(Y) = (1-t)c, \sigma(Y) = t$ 。根据 z_t 我们可以估计出 c ，其中估计值 $\hat{c} = \frac{1}{1-t}\frac{1}{n}\sum_{i=1}^{n} z_{t,i}$ ，

其标准差为 $\sigma(t,n) = \frac{t}{1-t}\sqrt{\frac{1}{n}}$ 。这里的标准差可以看成我们对 c 的破坏程度，可以看到当图像的宽和高都增大一倍时，破坏程度也相应降低了一倍。这里我们希望，分辨率 n 下的 $\sigma(t_n, n)$ 和分辨率 m 下的 $\sigma(t_m, m)$ 相同。求解可以得到：

$$t_m = \frac{\sqrt{\frac{m}{n}} t_n}{1 + (\sqrt{\frac{m}{n}} - 1) t_n}$$

根据上式，我们可以计算出SNR，有：

$$\lambda_{t_m} = 2 \log \frac{1 - t_m}{t_m} = 2 \log \frac{1 - t_n}{\sqrt{\frac{m}{n}} t_n} = \lambda_{t_n} - \log \frac{m}{n}$$

这意味两者的SNR要偏移一个 $\log \frac{m}{n}$ 。当分辨率变成1024x1024，论文中是通过人工评测实验来选择最优的 $\sqrt{\frac{m}{n}}$ ，实验最优值是3.0。

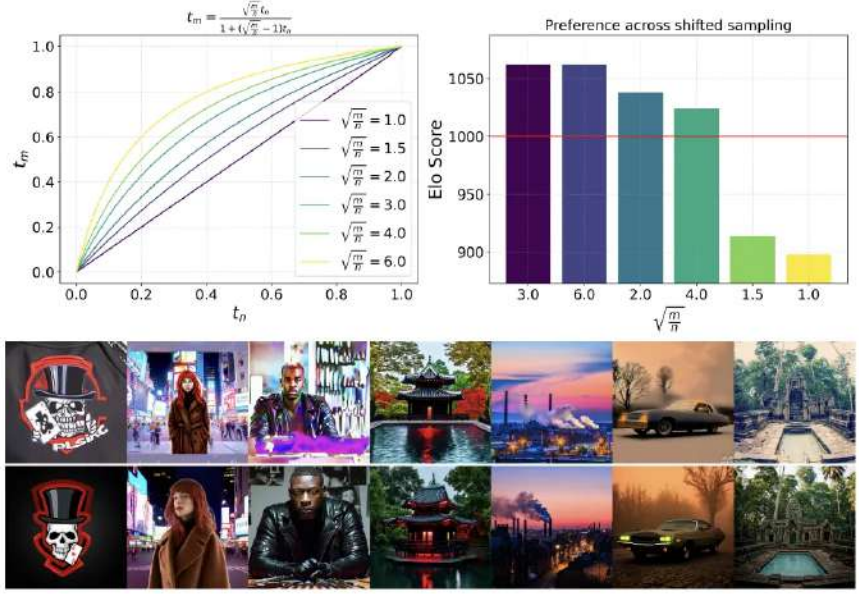


Figure 6. **Timestep shifting at higher resolutions.** *Top right:* Human quality preference rating when applying the shifting based on Equation (23). *Bottom row:* A 512^2 model trained and sampled with $\sqrt{m/n} = 1.0$ (*top*) and $\sqrt{m/n} = 3.0$ (*bottom*). See Section 5.3.2.

模型scaling

transformer一个比较大的优势是有好的scaling能力：当增大模型带来性能的稳定提升。论文中也选择了不同规模大小的MM-DiT进行实验，不同大小的网络深度分别是15，18，21，30，38，其中最大的模型参数量为8B。结论是MM-DiT同样表现了比较好的scaling能力，当模型变大后，性能稳步提升，如下图所示：

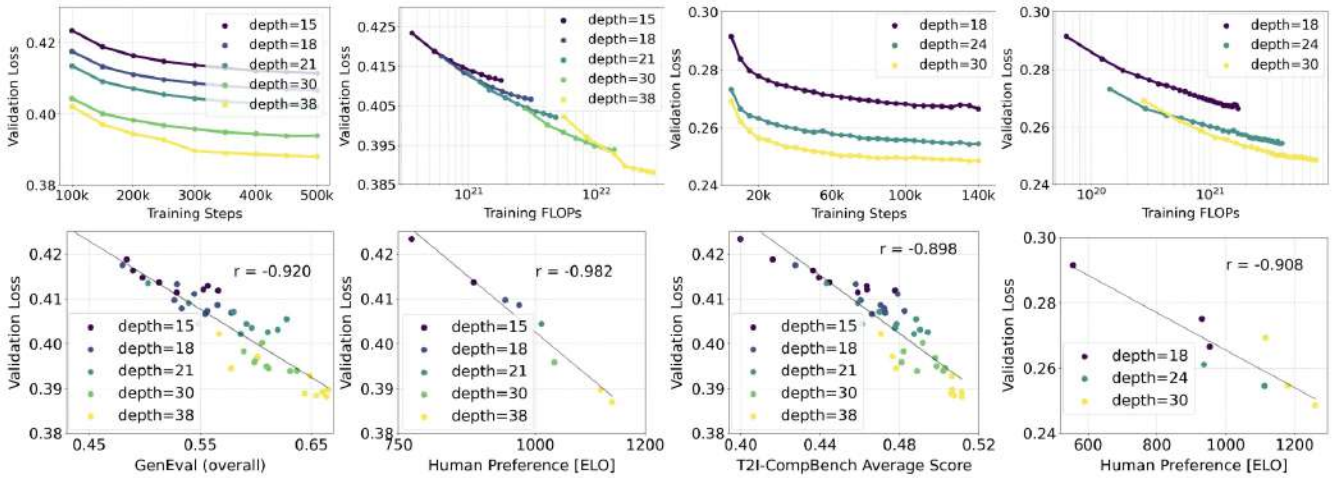


Figure 8. **Quantitative effects of scaling.** We analyze the impact of model size on performance, maintaining consistent training hyperparameters throughout. An exception is depth=38, where learning rate adjustments at 3×10^5 steps were necessary to prevent divergence. (Top) Validation loss smoothly decreases as a function of both model size and training steps for both image (columns 1 and 2) and video models (columns 3 and 4). (Bottom) Validation loss is a *strong predictor of overall model performance*. There is a marked correlation between validation loss and holistic image evaluation metrics, including GenEval (Ghosh et al., 2023), column 1, human preference, column 2, and T2I-CompBench (Huang et al., 2023), column 3. For video models we observe a similar correlation between validation loss and human preference, column 4. .

这里的另外一个结论是validation loss可以作为一个很好的模型性能的衡量指标，它和文生图模型的一些评测指标如CompBench和GenEval，以及人类偏好是正相关的。而且从目前的实验结果来看，还没有看到出现性能的饱和，这意味着继续增大模型，依然有可能继续提升。

下图展示了三个不同大小的模型生成图像的差异，可以看到大模型确实是质量最好的。

Figure 12. **Qualitative effects of scaling.** Displayed are examples demonstrating the impact of scaling training steps (left to right: 50k, 200k, 350k, 500k) and model sizes (top to bottom: depth=15, 30, 38) on PartiPrompts, highlighting the influence of training duration and model complexity.

而且更大的模型不仅性能更好，而且生成时可以用较少的采样步数，比如当步数为5步时，大模型的性能下降要比小模型要低。

	relative CLIP score decrease [%]			
	5/50 steps	10/50 steps	20/50 steps	path length
depth=15	4.30	0.86	0.21	191.13
depth=30	3.59	0.70	0.24	187.96
depth=38	2.71	0.14	0.08	185.96

Table 6. **Impact of model size on sampling efficiency.** The table shows the relative performance decrease relative to CLIP scores evaluated using 50 sampling steps at a fixed seed. Larger models can be sampled using fewer steps, which we attribute to increased robustness and better fitting the straight-path objective of rectified flow models, resulting in shorter path lengths. Path length is calculated by summing up $\|v_\theta \cdot dt\|$ over 50 steps.

实现细节

这部分简单介绍一下SD3的一些实现细节，包括训练数据的处理以及训练参数等。

预训练数据处理

预训练数据集的大小和来源是没有的，但是预训练数据会进行一些筛选，包括：

1. 色情内容：使用NSFW检测模型来过滤。
2. 图像美学：使用评分系统移除预测分数较低的图像。
3. 重复内容：基于聚类的去重方法来移除训练数据中重复的图像，防止模型直接复制训练数据集中图像。（这部分策略附录部分很详细）

图像caption

和DALL-E 3一样，这里也对训练数据集中的图像生成高质量caption，这里使用的模型是多模态大模型CogVLM。训练过程中，使用50%的原始caption和50%的合成caption，使用合成caption能够提升模型性能，如下表所示。

	Original Captions	50/50 Mix
	success rate [%]	success rate [%]
Color Attribution	11.75	24.75
Colors	71.54	68.09
Position	6.50	18.00
Counting	33.44	41.56
Single Object	95.00	93.75
Two Objects	41.41	52.53
Overall score	43.27	49.78

Table 4. **Improved Captions.** Using a 50/50 mixing ratio of synthetic (via CogVLM (Wang et al., 2023)) and original captions improves text-to-image performance. Assessed via the GenEval (Ghosh et al., 2023) benchmark.

预计算图像和文本特征

位了减少训练过程中所需显存，这里预先计算好图像经过autoencoder编码得到的latent，以及文本对应的text embedding，特别是T5，可以节省接近20B的显存。同时预先计算好特征，也会节省一部分时间。

Model	Mem [GB]	FP [ms]	Storage [kB]	Delta [%]
VAE (Enc)	0.14	2.45	65.5	13.8
CLIP-L	0.49	0.45	121.3	2.6
CLIP-G	2.78	2.77	202.2	15.6
T5	19.05	17.46	630.7	98.3

Table 7. **Key figures for preencoding frozen input networks.** Mem is the memory required to load the model on the GPU. FP [ms] is the time per sample for the forward pass with per-device batch size of 32. Storage is the size to save a single sample. Delta [%] is how much longer a training step takes, when adding this into the loop for the 2B MMDiT-Model (568ms/it).

但是预计算特征也不是没有代价的，首先是图像就不能做数据增强，好在文生图模型训练一般不太需要数据增强，其次需要一定的存储空间，而且加载特征也需要时间。预计算特征其实就是空间换时间。

Classifier-Free Guidance

训练过程需要对文本进行一定的drop来实现Classifier-Free Guidance，这里是三个text encoder各以46.4%的比例单独drop，这也意味着text完全drop的比例为 $(46.4\%)^3 \approx 10\%$ 。三个text encoder独立drop的一个好处是推理时可以灵活使用text encoder。比如，我们可以去掉比较吃显存的T5模型，只保留两个CLIP text encoder，实验发现这并不会影响视觉美感（没有T5的胜率为50%），并且只会导致

文本遵循度略有下降（胜率为46%），这种情况包括文本提示词包含高度详细的场景描述或大量文字。然而，如果想生成文字，还是加上T5，没有T5的胜率只有38%。下面是一些具体的例子：

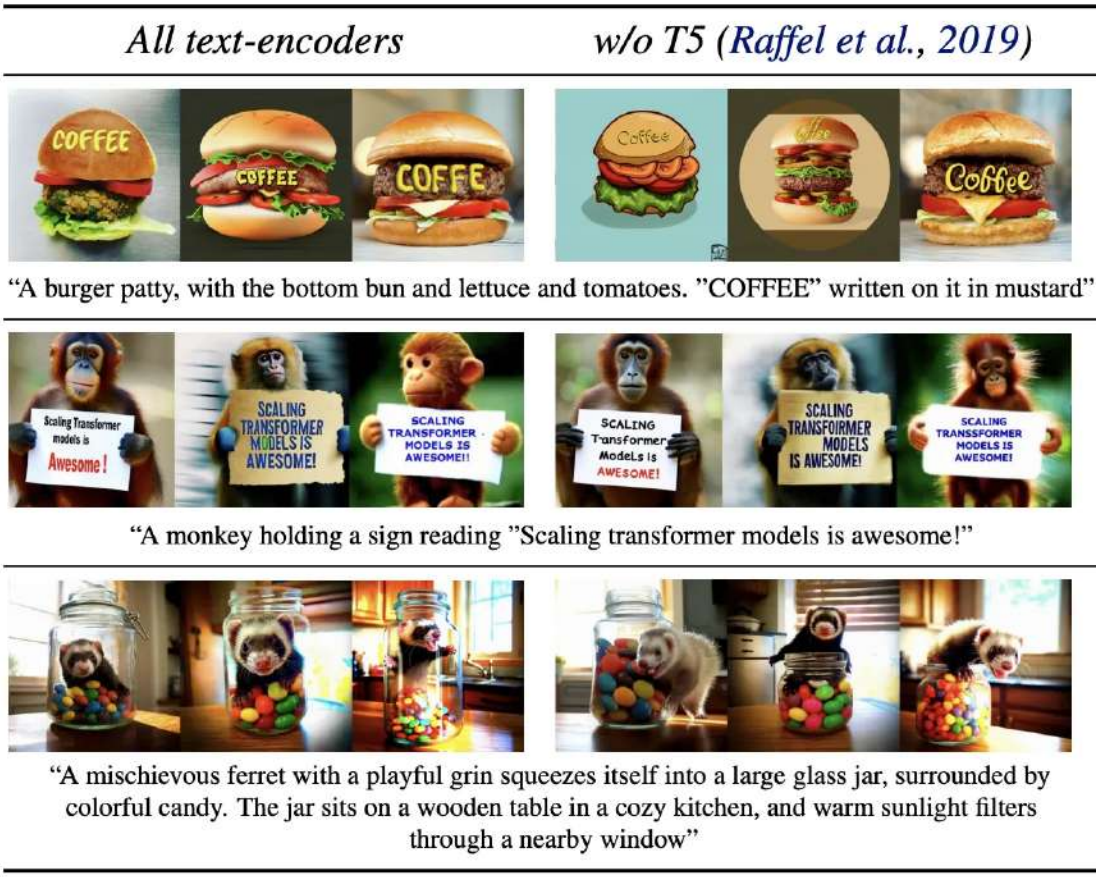


Figure 9. **Impact of T5.** We observe T5 to be important for complex prompts e.g. such involving a high degree of detail or longer spelled text (rows 2 and 3). For most prompts, however, we find that removing T5 at inference time still achieves competitive performance.

DPO

SD3最后基于DPO来进一步提升性能，DPO相比RLHF的一个优势不需要单独训练一个reward模型，而且直接基于成对的比较数据训练。DPO目前已经成功应用在文生图上：Diffusion Model Alignment Using Direct Preference Optimization。SD3这里没有finetune整个网络，而是基于rank=128的LoRA，经过DPO后，图像生成质量有一定的提升，如下所示：

Figure 13. Comparison between base models and DPO-finetuned models. DPO-finetuning generally results in more aesthetically pleasing samples with better spelling.

性能评测

性能评测包括定量评测和人工评测。

定量评测

定量评测基于GenEval，SD3和其它模型的对比如下所示，可以看到最大的模型在经过DPO后超过DALL-E 3。

Model	Overall	Objects		Counting	Colors	Position	Color Attribution
		Single	Two				
minDALL-E	0.23	0.73	0.11	0.12	0.37	0.02	0.01
SD v1.5	0.43	0.97	0.38	0.35	0.76	0.04	0.06
PixArt-alpha	0.48	0.98	0.50	0.44	0.80	0.08	0.07
SD v2.1	0.50	0.98	0.51	0.44	0.85	0.07	0.17
DALL-E 2	0.52	0.94	0.66	0.49	0.77	0.10	0.19
SDXL	0.55	0.98	0.74	0.39	0.85	0.15	0.23
SDXL Turbo	0.55	**1.00**	0.72	0.49	0.80	0.10	0.18
IF-XL	0.61	0.97	0.74	*0.66*	0.81	0.13	0.35
DALL-E 3	0.67	0.96	0.87	0.47	*0.83*	**0.43**	*0.45*
Ours (depth=18), 512^2	0.58	0.97	0.72	0.52	0.78	0.16	0.34
Ours (depth=24), 512^2	0.62	0.98	0.74	0.63	0.67	*0.34*	0.36
Ours (depth=30), 512^2	0.64	0.96	0.80	0.65	0.73	0.33	0.37
Ours (depth=38), 512^2	*0.68*	0.98	0.84	*0.66*	0.74	0.40	0.43
Ours (depth=38), 512^2 w/DPO	0.71	0.98	0.89	**0.73**	*0.83*	*0.34*	0.47
Ours (depth=38), 1024^2 w/DPO	**0.74**	0.99	**0.94**	0.72	**0.89**	0.33	**0.60**

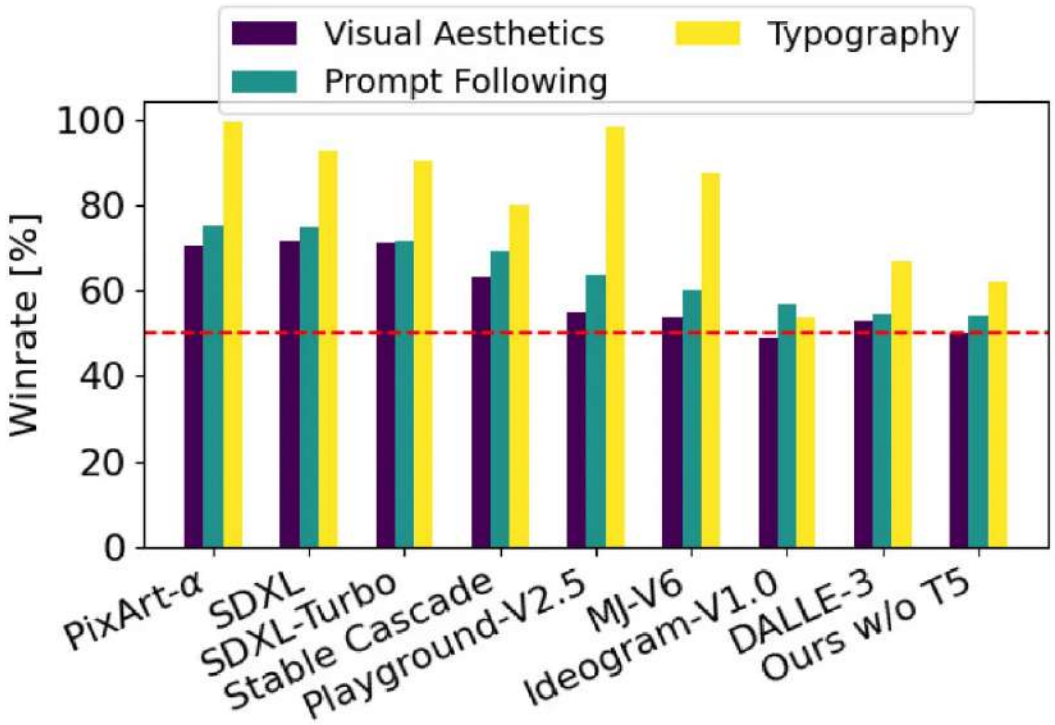
Table 5. **GenEval comparisons**. Our largest model (depth=38) outperforms all current open models and DALLE-3 (Betker et al., 2023) on GenEval (Ghosh et al., 2023). We highlight the **best**, second best, and *third best* entries. For DPO, see Appendix C.

人工评测

人工评测包括三个方面：

> **Prompt following**: Which image looks more representative to the text shown above and faithfully follows it?
>
> **Visual aesthetics**: Given the prompt, which image is of higher-quality and aesthetically more pleasing?
>
> **Typography**: Which image more accurately shows/displays the text specified in the above description? More accurate spelling is preferred! Ignore other aspects.

评测结果如下所示，这里对比的模型有SOTA的模型：MJ-V6，Ideogram-V1.0，DALL-E 3，在文字生成方面，SD3基本大幅赢过其它模型（和Ideogram-V1.0相差上下），在图像质量和文本提示词遵循方面也和SOTA模型不相上下。



小结

SD3可以说是集大成者，基本上把业界最好的或者最成熟的方案都用上了，比如RF和DiT，以及DPO等等。SD3的正式发布，也基本宣告文生图进入transformer时代了，现在的模型才是8B，未来更大的模型也定会出现。

参考

- https://stability.ai/news/stable-diffusion-3-research-paper
- https://arxiv.org/abs/2212.09748
- https://arxiv.org/abs/2403.03206
- https://arxiv.org/abs/2210.02747
- https://arxiv.org/abs/2303.00848
- https://arxiv.org/abs/2209.03003
- https://arxiv.org/abs/2209.14577