

面向高分辨率图像合成的修正流 Transformer 缩放研究

Scaling Rectified Flow Transformers for High-Resolution Image Synthesis

Patrick Esser* Sumith Kulal Andreas Blattmann Rahim Entezari Jonas Müller
Harry Saini Yam Levi Dominik Lorenz Axel Sauer Frederic Boesel Dustin Podell
Tim Dockhorn Zion English
Kyle Lacey Alex Goodwin Yannik Marek Robin Rombach*
Stability AI



图 1. High-resolution samples from our 8B rectified flow model, showcasing its capabilities in typography, precise prompt following and spatial reasoning, attention to fine details, and high image quality across a wide variety of styles. 这些高分辨率样本由我们的 80 亿参数修正流模型生成，展现出它在多方面的能力：文字排版、精确的提示遵循与空间推理、对精细细节的刻画，以及在各类风格下都能保持高图像质量。

摘要

Diffusion models create data from noise by inverting the forward paths of data towards noise and have emerged as a powerful generative modeling

*Equal contribution . <first.last>@stability.ai.

本文为 AI 辅助翻译版本，仅供学习交流；引用请以英文原文为准：arXiv:2403.03206。

technique for high-dimensional, perceptual data such as images and videos. Rectified flow is a recent generative model formulation that connects data and noise in a straight line. Despite its better theoretical properties and conceptual simplicity, it is not yet decisively established as standard practice. In this work, we improve existing noise sampling techniques for training rectified flow models by biasing them towards perceptually relevant scales. Through a large-scale study, we

demonstrate the superior performance of this approach compared to established diffusion formulations for high-resolution text-to-image synthesis. Additionally, we present a novel transformer-based architecture for text-to-image generation that uses separate weights for the two modalities and enables a bidirectional flow of information between image and text tokens, improving text comprehension, typography, and human preference ratings. We demonstrate that this architecture follows predictable scaling trends and correlates lower validation loss to improved text-to-image synthesis as measured by various metrics and human evaluations. Our largest models outperform state-of-the-art models, and we will make our experimental data, code, and model weights publicly available.

扩散模型通过反演前向路径从噪声生成数据,是处理高维感知数据(如图像和视频)的强大生成建模方法。修正流(Rectified Flow, RF)是一种最近的生成模型公式,通过直线连接数据和噪声。尽管理论特性更优、概念更简洁,修正流尚未被广泛采纳。本工作通过将噪声采样偏向感知相关的尺度,改进了训练修正流模型的现有采样技术。大规模研究表明,该方法相比现有扩散公式在高分辨率文本到图像合成中具有优越性能。我们提出一种新的基于 Transformer 的文本到图像架构,为两个模态分别使用权重,建立图像和文本 token 间的双向信息流,改进了提示理解、文字排版和人类偏好评分。该架构遵循可预测的缩放趋势,验证损失降低与文本到图像合成质量提升相关(用多种指标和人类评估验证)。最大的模型超越最先进的现有模型,我们公开代码、模型和实验数据。

1. 引言

Diffusion models create data from noise (Song 等, 2020). They are trained to

扩散模型从噪声中生成数据 (Song 等, 2020)。在训练过程中,它们学会 invert forward paths of data towards random noise and, thus, in conjunction with approximation and generalization properties of neural networks, can

be used to generate new data points that are not present in the training data but follow the distribution of the training data (Sohl-Dickstein 等, 2015; Song & Ermon, 2020). This generative modeling technique has proven to be very effective for modeling high-dimensional, perceptual data such as images (Ho 等, 2020). In recent years, diffusion models have become the de-facto approach for generating high-resolution images and videos from natural language inputs with impressive generalization capabilities (Saharia 等, 2022b; Ramesh 等, 2022; Rombach 等, 2022; Podell 等, 2023; Dai 等, 2023; Esser 等, 2023; Blattmann 等, 2023b; Betker 等, 2023; Blattmann 等, 2023a; Singer 等, 2022). Due to their iterative nature and the associated computational costs, as well as the long sampling times during inference, research on formulations for more efficient training and/or faster sampling of these models has increased (Karras 等, 2023; Liu 等, 2022).

反演从数据到随机噪声的前向路径。结合神经网络的近似和泛化能力,扩散模型可用于生成训练数据中不存在但遵循训练数据分布的新数据点 (Sohl-Dickstein 等, 2015; Song & Ermon, 2020)。这种生成建模技术已被证明非常有效于建模高维感知数据(如图像) (Ho 等, 2020)。近年来,扩散模型已成为从自然语言输入生成高分辨率图像和视频的实际标准方法,具有令人印象深刻的泛化能力 (Saharia 等, 2022b; Ramesh 等, 2022; Rombach 等, 2022; Podell 等, 2023; Dai 等, 2023; Esser 等, 2023; Blattmann 等, 2023b; Betker 等, 2023; Blattmann 等, 2023a; Singer 等, 2022)。由于这些模型的迭代性质和相关的计算成本,以及推理过程中漫长的采样时间,关于更高效训练和/或更快采样公式的研究不断增加 (Karras 等, 2023; Liu 等, 2022)。

While specifying a forward path from data to noise leads to efficient training, it also raises the question of which path to choose. This choice can have important implications for sampling. For example, a forward process that fails to remove all noise from the data can lead to a discrepancy in training and test distribution and result in artifacts such as gray image samples (Lin 等, 2024). Importantly, the choice of the forward process also influences the learned backward process and, thus, the sampling efficiency. While curved paths require many integration steps to simulate the process, a straight path could be simulated with a single step and is less prone to error accumulation. Since each step corresponds to an evaluation of the neural

network, this has a direct impact on the sampling speed.

虽然指定从数据到噪声的前向路径可以实现高效训练，但也引发了选择哪条路径的问题。这一选择对采样可能具有重要影响。例如，无法彻底去除数据中所有噪声的前向过程可能导致训练和测试分布之间的偏差，并产生如灰色图像样本等伪影 (Lin 等, 2024)。重要的是，前向过程的选择也会影响学到的反向过程，进而影响采样效率。曲线路径需要多个积分步骤才能模拟该过程，而直线路径只需一步即可模拟，且不易产生误差积累。由于每一步都对应神经网络的一次评估，这直接影响采样速度。

A particular choice for the forward path is a so-called *Rectified Flow* (Liu 等, 2022; Albergo & Vanden-Eijnden, 2022; Lipman 等, 2023), which connects data and noise on a straight line. Although this model class has better theoretical properties, it has not yet become decisively established in practice. So far, some advantages have been empirically demonstrated in small and medium-sized experiments (Ma 等, 2024), but these are mostly limited to class-conditional models. In this work, we change this by introducing a re-weighting of the noise scales in rectified flow models, similar to noise-predictive diffusion models (Ho 等, 2020). Through a large-scale study, we compare our new formulation to existing diffusion formulations and demonstrate its benefits.

一种特定的前向路径选择是所谓的修正流 (*Rectified Flow*, *RF*) (Liu 等, 2022; Albergo & Vanden-Eijnden, 2022; Lipman 等, 2023)，它在直线上连接数据和噪声。虽然这个模型类具有更好的理论性质，但在实践中还未被广泛采用。一些优势已在小规模和中等规模实验中得到验证 (Ma 等, 2024)，但这些主要局限于类条件模型。在本工作中，我们通过重新加权修正流模型中的噪声尺度来改变这一现状，这类类似于噪声预测扩散模型 (Ho 等, 2020)。通过大规模研究，我们将新的公式与现有扩散公式进行比较，并证明其优势。

We show that the widely used approach for text-to-image synthesis, where a fixed text representation is fed directly into the model (e.g., via cross-attention (Vaswani 等, 2017; Rombach 等, 2022)), is not ideal, and present a new architecture that incorporates learnable streams for both image and text tokens, which enables a two-way flow

of information between them. We combine this with our improved rectified flow formulation and investigate its scalability. We demonstrate a predictable scaling trend in the validation loss and show that a lower validation loss correlates strongly with improved automatic and human evaluations.

我们表明，文本到图像合成中广泛采用的方法——将固定文本表示直接输入模型（例如通过交叉注意力 (Vaswani 等, 2017; Rombach 等, 2022)）——效果并非最优。我们提出了一个新架构，为图像和文本 token 分别设计可学习的流，使二者能够相互交互。我们将这一点与改进的修正流公式相结合，并研究其可扩展性。我们展示了验证损失的可预测缩放趋势，并表明较低的验证损失与更好的自动评估和人类评估密切相关。

Our largest models outperform state-of-the-art open models such as *SDXL* (Podell 等, 2023), *SDXL-Turbo* (Sauer 等, 2023), *Pixart- α* (Chen 等, 2023), and closed-source models such as DALL-E 3 (Betker 等, 2023) both in quantitative evaluation (Ghosh 等, 2023) of prompt understanding and human preference ratings.

我们最大的模型在提示理解的定量评估 (Ghosh 等, 2023) 和人类偏好评分上都超越了开源的 *SDXL* (Podell 等, 2023)、*SDXL-Turbo* (Sauer 等, 2023)、*Pixart- α* (Chen 等, 2023) 以及闭源的 DALL-E 3 (Betker 等, 2023)。

The core contributions of our work are: (i) We conduct a large-scale, systematic study on different diffusion model and rectified flow formulations to identify the best setting. For this purpose, we introduce new noise samplers for rectified flow models that improve performance over previously known samplers. (ii) We devise a novel, scalable architecture for text-to-image synthesis that allows bi-directional mixing between text and image token streams within the network. We show its benefits compared to established backbones such as UViT (Hoogeboom 等, 2023) and DiT (Peebles & Xie, 2023). Finally, we (iii) perform a scaling study of our model and demonstrate that it follows predictable scaling trends. We show that a lower validation loss correlates strongly with improved text-to-image performance assessed via metrics such as T2I-CompBench (Huang 等, 2023), GenEval (Ghosh 等, 2023) and human ratings.

本工作的核心贡献如下：(i) 我们对不同的扩散模型和修正流公式进行了大规模系统性研究，以确定最优设置。为此，我们为修正流模型引入了新的噪声采样器，相比之前已知的采样器性能有所改进。(ii) 我们设计了一个可扩展的新型文本到图像合成架构，使文本和图像 token 流在网络中能够双向交互。我们证明了相比现有的骨干网络（如 UViT (Hoogeboom 等, 2023) 和 DiT (Peebles & Xie, 2023)）的优势。最后，我们 (iii) 对模型进行了缩放研究，并展示了它遵循可预测的缩放趋势。我们表明，较低的验证损失与通过 T2I-CompBench (Huang 等, 2023)、GenEval (Ghosh 等, 2023) 等指标和人类评分评估的改进的文本到图像性能密切相关。We make results, code, and model weights publicly available.

我们公开发布结果、代码和模型权重。

2. 免仿真流训练

We consider generative models that define a mapping between samples x_1 from a noise distribution p_1 to samples x_0 from a data distribution p_0 in terms of an ordinary differential equation (ODE),

我们考虑生成模型，将噪声分布 p_1 中的样本 x_1 映射到数据分布 p_0 中的样本 x_0 ，用常微分方程 (ODE) 表示，

$$dy_t = v_\Theta(y_t, t) dt, \quad (1)$$

where the velocity v is parameterized by the weights Θ of a neural network. Prior work by Chen 等 (2018) suggested to directly solve Equation (1) via differentiable ODE solvers. However, this process is computationally expensive, especially for large network architectures that parameterize $v_\Theta(y_t, t)$. A more efficient alternative is to directly regress a vector field u_t that generates a probability path between p_0 and p_1 .

其中速度 v 由神经网络权重 Θ 参数化。之前 Chen 等 (2018) 的工作建议通过可微分 ODE 求解器直接求解 Equation (1)。但这个过程计算成本很高，特别是对于参数化 $v_\Theta(y_t, t)$ 的大型网络架构。更有效的替代方案是直接回归向量场 u_t ，用其生成 p_0 和 p_1 之间的概率路径。To construct such a u_t , we define a forward process, corresponding to a probability path p_t between p_0 and $p_1 = \mathcal{N}(0, 1)$, as

为了构造这样的 u_t ，我们定义前向过程，对应于 p_0 和 $p_1 = \mathcal{N}(0, 1)$ 之间的概率路径 p_t ，如下

$$z_t = a_t x_0 + b_t \epsilon \quad \text{where } \epsilon \sim \mathcal{N}(0, I). \quad (2)$$

For $a_0 = 1, b_0 = 0, a_1 = 0$ and $b_1 = 1$, the marginals,

对于 $a_0 = 1, b_0 = 0, a_1 = 0$ 和 $b_1 = 1$ ，边缘分布，

$$p_t(z_t) = \mathbb{E}_{\epsilon \sim \mathcal{N}(0, I)} p_t(z_t | \epsilon), \quad (3)$$

are consistent with the data and noise distribution.

与数据分布和噪声分布一致。

To express the relationship between z_t, x_0 and ϵ , we introduce

ψ_t and u_t as

为了表达 z_t, x_0 和 ϵ 之间的关系，我们引入 ψ_t 和 u_t 如下

$$\psi_t(\cdot | \epsilon) : x_0 \mapsto a_t x_0 + b_t \epsilon \quad (4)$$

$$u_t(z | \epsilon) := \psi_t'(\psi_t^{-1}(z | \epsilon) | \epsilon) \quad (5)$$

Since z_t can be written as solution to the ODE $z_t' = u_t(z_t | \epsilon)$, with initial value $z_0 = x_0$, $u_t(\cdot | \epsilon)$ generates $p_t(\cdot | \epsilon)$.

由于 z_t 可以写成 ODE $z_t' = u_t(z_t | \epsilon)$ 的解，初始值为 $z_0 = x_0$ ， $u_t(\cdot | \epsilon)$ 生成 $p_t(\cdot | \epsilon)$ 。Remarkably, one can construct a marginal vector field u_t which generates the marginal probability paths p_t (Lipman 等, 2023) (see B.1), using the conditional vector fields $u_t(\cdot | \epsilon)$:

可以利用条件向量场 $u_t(\cdot | \epsilon)$ 构造边缘向量场 u_t ，它生成边缘概率路径 p_t (Lipman 等, 2023) (见 B.1):

$$u_t(z) = \mathbb{E}_{\epsilon \sim \mathcal{N}(0, I)} u_t(z | \epsilon) \frac{p_t(z | \epsilon)}{p_t(z)} \quad (6)$$

虽然使用流匹配目标直接回归 u_t 由于 Equation 6 中的边缘化而难以求解，但条件流匹配 (见 B.1)

$$\mathcal{L}_{CFM} = \mathbb{E}_{t, p_t(z | \epsilon), p(\epsilon)} \|v_\Theta(z, t) - u_t(z | \epsilon)\|_2^2, \quad (7)$$

with the conditional vector fields $u_t(z | \epsilon)$ provides an equivalent yet tractable objective.

使用条件向量场 $u_t(z | \epsilon)$ 提供了等价但可求解的目标。

To convert the loss into an explicit form we insert $\psi_t'(x_0 | \epsilon) = a_t' x_0 + b_t' \epsilon$ and $\psi_t^{-1}(z | \epsilon) = \frac{z - b_t \epsilon}{a_t}$ into (5)

为了将损失转换为显式形式，我们将 $\psi_t'(x_0 | \epsilon) = a_t' x_0 + b_t' \epsilon$ 和 $\psi_t^{-1}(z | \epsilon) = \frac{z - b_t \epsilon}{a_t}$ 代入 (5)

$$z_t' = u_t(z_t | \epsilon) = \frac{a_t'}{a_t} z_t - \epsilon b_t \left(\frac{a_t'}{a_t} - \frac{b_t'}{b_t} \right). \quad (8)$$

Now, consider the *signal-to-noise ratio* $\lambda_t := \log \frac{a_t^2}{b_t^2}$. With $\lambda'_t = 2(\frac{a'_t}{a_t} - \frac{b'_t}{b_t})$, we can rewrite Equation (8) as

现在考虑 信噪比 $\lambda_t := \log \frac{a_t^2}{b_t^2}$ 。利用 $\lambda'_t = 2(\frac{a'_t}{a_t} - \frac{b'_t}{b_t})$ ，我们可以将 Equation (8) 改写为

$$u_t(z_t|\epsilon) = \frac{a'_t}{a_t} z_t - \frac{b_t}{2} \lambda'_t \epsilon \quad (9)$$

Next, we use Equation (9) to reparameterize Equation (7) as a noise-prediction objective:

接下来，我们使用 Equation (9) 将 Equation (7) 重新参数化为噪声预测目标：

$$\mathcal{L}_{CFM} = \mathbb{E}_{t, p_t(z|\epsilon), p(\epsilon)} \|v_\Theta(z, t) - \frac{a'_t}{a_t} z + \frac{b_t}{2} \lambda'_t \epsilon\|_2^2 \quad (10)$$

$$= \mathbb{E}_{t, p_t(z|\epsilon), p(\epsilon)} \left(-\frac{b_t}{2} \lambda'_t \right)^2 \|\epsilon_\Theta(z, t) - \epsilon\|_2^2 \quad (11)$$

where we defined $\epsilon_\Theta := \frac{-2}{\lambda'_t b_t} (v_\Theta - \frac{a'_t}{a_t} z)$.

其中我们定义了 $\epsilon_\Theta := \frac{-2}{\lambda'_t b_t} (v_\Theta - \frac{a'_t}{a_t} z)$ 。

Note that the optimum of the above objective does not change when introducing a time-dependent weighting. Thus, one can derive various weighted loss functions that provide a signal towards the desired solution but might affect the optimization trajectory. For a unified analysis of different approaches, including classic diffusion formulations, we can write the objective in the following form (following Kingma & Gao (2023)):

注意，上述目标的最优值在引入时间相关的加权时不会改变。因此，可以导出各种加权损失函数，为所需的解提供信号，但可能会影响优化轨迹。为了统一分析不同的方法（包括经典扩散公式），我们可以按以下形式写出目标（遵循 Kingma & Gao (2023)）：

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

where $w_t = -\frac{1}{2} \lambda'_t b_t^2$ corresponds to $\mathcal{L}_{CFM}$.

其中 $w_t = -\frac{1}{2} \lambda'_t b_t^2$ 对应于 $\mathcal{L}_{CFM}$ 。

3. 流轨迹

In this work, we consider different variants of the above formalism that we briefly describe in the following.

在本工作中，我们考虑上述形式的不同变体，简要描述如下。

Rectified Flow Rectified Flows (RFs) (Liu 等, 2022; Albergo & Vanden-Eijnden, 2022; Lipman 等, 2023) define the forward process as straight paths between the data distribution and a standard normal distribution, i.e.

修正流 修正流 (Rectified Flow, RF) (Liu 等, 2022; Albergo & Vanden-Eijnden, 2022; Lipman 等, 2023) 定义前向过程为数据分布和标准正态分布之间的直线路径，即

$$z_t = (1-t)x_0 + t\epsilon, \quad (12)$$

and uses $\mathcal{L}_{CFM}$ which then corresponds to $w_t^{\text{RF}} = \frac{t}{1-t}$. The network output directly parameterizes the velocity v_Θ .

使用 $\mathcal{L}_{CFM}$ ，对应于 $w_t^{\text{RF}} = \frac{t}{1-t}$ 。网络输出直接参数化速度 v_Θ 。

EDM EDM (Karras 等, 2022) uses a forward process of the form

EDM (Karras 等, 2022) 使用如下形式的前向过程

$$z_t = x_0 + b_t \epsilon \quad (13)$$

其中 (Kingma & Gao, 2023) $b_t = \exp F_{\mathcal{N}}^{-1}(t|P_m, P_s^2)$ with $F_{\mathcal{N}}^{-1}$ being the quantile function of the normal distribution with mean P_m and variance P_s^2 . Note that this choice results in

$F_{\mathcal{N}}^{-1}$ 是均值为 P_m 、方差为 P_s^2 的正态分布的分位数函数。注意这个选择导致

$$\lambda_t \sim \mathcal{N}(-2P_m, (2P_s)^2) \quad \text{for } t \sim \mathcal{U}(0, 1) \quad (14)$$

The network is parameterized through an $\mathbf{F}$ -prediction (Kingma & Gao, 2023; Karras 等, 2022) and the loss can be written as $\mathcal{L}_{w_t^{\text{EDM}}}$ with

网络通过 $\mathbf{F}$ 预测参数化 (Kingma & Gao, 2023; Karras 等, 2022)，损失可以写成 $\mathcal{L}_{w_t^{\text{EDM}}}$ ，其中

$$w_t^{\text{EDM}} = \mathcal{N}(\lambda_t | -2P_m, (2P_s)^2) (e^{-\lambda_t} + 0.5^2) \quad (15)$$

Cosine (Nichol & Dhariwal, 2021) proposed a forward process of the form

余弦 (Nichol & Dhariwal, 2021) 提出如下形式的前向过程

$$z_t = \cos\left(\frac{\pi}{2}t\right)x_0 + \sin\left(\frac{\pi}{2}t\right)\epsilon. \quad (16)$$

In combination with an ϵ -parameterization and loss, this corresponds to a weighting $w_t = \text{sech}(\lambda_t/2)$. When combined with a $\mathbf{v}$ -prediction loss (Kingma & Gao, 2023), the weighting is given by $w_t = e^{-\lambda_t/2}$.

结合 ϵ 参数化和损失, 这对应于加权 $w_t = \text{sech}(\lambda_t/2)$ 。当与 $\mathbf{v}$ 预测损失 (Kingma & Gao, 2023) 结合时, 加权由 $w_t = e^{-\lambda_t/2}$ 给出。

(LDM-)Linear LDM (Rombach 等, 2022) uses a modification of the DDPM schedule (Ho 等, 2020). Both are variance preserving schedules, i.e. $b_t = \sqrt{1 - a_t^2}$, and define a_t for discrete timesteps $t = 0, \dots, T-1$ in terms of diffusion coefficients β_t as

(LDM-) 线性 LDM (Rombach 等, 2022) 使用 DDPM 调度的修改 (Ho 等, 2020)。两者都是方差保留的调度, 即 $b_t = \sqrt{1 - a_t^2}$, 并用扩散系数 β_t 为离散时间步 $t = 0, \dots, T-1$ 定义 a_t , 如下 $a_t = (\prod_{s=0}^t (1 - \beta_s))^{1/2}$ 。For given boundary values β_0 and β_{T-1} , DDPM uses

对于给定的边界值 β_0 和 β_{T-1} , DDPM 使用 $\beta_t = \beta_0 + \frac{t}{T-1}(\beta_{T-1} - \beta_0)$ and LDM uses

LDM 使用 $\beta_t = \left(\sqrt{\beta_0} + \frac{t}{T-1}(\sqrt{\beta_{T-1}} - \sqrt{\beta_0})\right)^2$ 。

3.1. 为 RF 模型定制的 SNR 采样器

The RF loss trains the velocity v_Θ uniformly on all timesteps in $[0, 1]$. Intuitively, however, the resulting velocity prediction target $\epsilon - x_0$ is more difficult for t in the middle of $[0, 1]$, since for $t = 0$, the optimal prediction is the mean of p_1 , and for $t = 1$ the optimal prediction is the mean of p_0 . In general, changing the distribution over t from the commonly used uniform distribution $\mathcal{U}(t)$ to a distribution with density $\pi(t)$ is equivalent to a weighted loss $\mathcal{L}_{w_t^\pi}$ with

RF 损失在 $[0, 1]$ 中的所有时间步上均匀地训练速度 v_Θ 。但得到的速度预测目标 $\epsilon - x_0$ 对 $[0, 1]$ 中间的 t 来说更困难: 对于 $t = 0$, 最优预测是 p_1 的均值, 对于 $t = 1$, 最优预测是 p_0 的均值。一般来说, 将 t 的分布从常用的均匀分布 $\mathcal{U}(t)$ 改变为密度为 $\pi(t)$ 的分布等价于加权损失 $\mathcal{L}_{w_t^\pi}$, 其中

$$w_t^\pi = \frac{t}{1-t} \pi(t) \quad (17)$$

Thus, we aim to give more weight to intermediate timesteps by sampling them more frequently. Next, we describe the timestep densities $\pi(t)$ that we use to train our models.

因此, 我们的目标是通过更频繁地采样中间时间步来给它们更多的权重。接下来, 我们描述用来训练模型的时间步密度 $\pi(t)$ 。

Logit-Normal Sampling One option for a distribution that puts more weight on intermediate steps is the logit-normal distribution (Atchison & Shen, 1980). Its density,

Logit 正态采样 一个选项是 logit 正态分布 (Atchison & Shen, 1980), 它在中间步骤上放置更多权重。其密度为

$$\pi_{\text{ln}}(t; m, s) = \frac{1}{s\sqrt{2\pi}} \frac{1}{t(1-t)} \exp\left(-\frac{(\text{logit}(t) - m)^2}{2s^2}\right), \quad (18)$$

where $\text{logit}(t) = \log \frac{t}{1-t}$, has a location parameter, m , and a scale parameter, s . The location parameter enables us to bias the training timesteps towards either data p_0 (negative m) or noise p_1 (positive m). As shown in Figure 11, the scale parameters controls how wide the distribution is.

其中 $\text{logit}(t) = \log \frac{t}{1-t}$, 具有位置参数 m 和尺度参数 s 。位置参数使我们能够将训练时间步偏置到数据 p_0 (负的 m) 或噪声 p_1 (正的 m)。如 Figure 11 所示, 尺度参数控制分布的宽度。

In practice, we sample the random variable u from a normal distribution $u \sim \mathcal{N}(u; m, s)$ and map it through the standard logistic function.

在实践中, 我们从正态分布 $u \sim \mathcal{N}(u; m, s)$ 中采样随机变量 u , 并通过标准逻辑函数映射它。

Mode Sampling with Heavy Tails

具有重尾的众数采样 The logit-normal density always vanishes at the endpoints 0 and 1. To study whether this has adverse effects on the performance, we also use a timestep sampling distribution with strictly positive density on $[0, 1]$. For a scale parameter s , we define

Logit 正态密度在端点 0 和 1 处总是消失。为了研究这是否对性能有不利影响, 我们也使用在 $[0, 1]$ 上具有严格正密度的时间步采样分布。对于尺度参数 s , 我们定义

$$f_{\text{mode}}(u; s) = 1 - u - s \cdot \left(\cos^2\left(\frac{\pi}{2}u\right) - 1 + u\right). \quad (19)$$

For $-1 \leq s \leq \frac{2}{\pi-2}$, this function is monotonic, and we

can use it to sample from the implied density $\pi_{\text{mode}}(t; s) = \left| \frac{d}{dt} f_{\text{mode}}^{-1}(t) \right|$.

对于 $-1 \leq s \leq \frac{2}{\pi-2}$, 这个函数是单调的, 我们可以用它来从隐含的密度 $\pi_{\text{mode}}(t; s) = \left| \frac{d}{dt} f_{\text{mode}}^{-1}(t) \right|$ 中采样。As seen in Figure 11, the scale parameter controls the degree to which either the midpoint (positive s) or the endpoints (negative s) are favored during sampling. This formulation also includes a **uniform weighting** $\pi_{\text{mode}}(t; s = 0) = \mathcal{U}(t)$ for $s = 0$, which has been used widely in previous works on Rectified Flows (Liu 等, 2022; Ma 等, 2024).

如 Figure 11 所示, 尺度参数控制在采样过程中中点 (正的 s) 或端点 (负的 s) 被偏好的程度。这个公式还包括 **均匀加权** $\pi_{\text{mode}}(t; s = 0) = \mathcal{U}(t)$ ($s = 0$ 时), 在之前关于修正流的工作中被广泛使用 (Liu 等, 2022; Ma 等, 2024)。

CosMap Finally, we also consider the *cosine* schedule (Nichol & Dhariwal, 2021) from Section 3 in the RF setting. In particular, we are looking for a mapping $f: u \mapsto f(u) = t$, $u \in [0, 1]$, such that the log-snr matches that of the cosine schedule: $2 \log \frac{\cos(\frac{\pi}{2}u)}{\sin(\frac{\pi}{2}u)} = 2 \log \frac{1-f(u)}{f(u)}$. Solving for f , we obtain for $u \sim \mathcal{U}(u)$

余弦映射 最后, 我们也在 RF 设置中考虑 Section 3 中的 余弦调度 (Nichol & Dhariwal, 2021)。特别地, 我们寻找映射 $f: u \mapsto f(u) = t$, $u \in [0, 1]$, 使得 log-snr 与余弦调度匹配: $2 \log \frac{\cos(\frac{\pi}{2}u)}{\sin(\frac{\pi}{2}u)} = 2 \log \frac{1-f(u)}{f(u)}$ 。求解 f , 对于 $u \sim \mathcal{U}(u)$ 我们得到

$$t = f(u) = 1 - \frac{1}{\tan(\frac{\pi}{2}u) + 1}, \quad (20)$$

from which we obtain the density

从中我们得到密度

$$\pi_{\text{CosMap}}(t) = \left| \frac{d}{dt} f^{-1}(t) \right| = \frac{2}{\pi - 2\pi t + 2\pi t^2}. \quad (21)$$

4. 文本到图像架构

For text-conditional sampling of images, our model has to take both modalities, text and images, into account. We use pretrained models to derive suitable representations and then describe the architecture of our diffusion backbone. An overview of this is presented in Figure 2.

对于图像的文本条件采样, 我们的模型需要考虑文本和图像这两种模态。我们使用预训练模型来获得

合适的表示, 然后描述我们扩散骨干网络的架构。这的一个概述在 Figure 2 中给出。

Our general setup follows LDM (Rombach 等, 2022) for training text-to-image models in the latent space of a pretrained autoencoder. Similar to the encoding of images to latent representations, we also follow previous approaches (Saharia 等, 2022b; Balaji 等, 2022) and encode the text conditioning c using pretrained, frozen text models. Details can be found in Appendix B.2.

我们的总体设置遵循 LDM (Rombach 等, 2022) 在预训练自编码器的潜在空间中训练文本到图像模型。类似于图像到潜在表示的编码, 我们也遵循之前的方法 (Saharia 等, 2022b; Balaji 等, 2022), 使用预训练的冻结文本模型编码文本条件 c 。详见 Appendix B.2。

Multimodal Diffusion Backbone

多模态扩散骨干网络 Our architecture builds upon the DiT (Peebles & Xie, 2023) architecture. DiT only considers class conditional image generation and uses a modulation mechanism to condition the network on both the timestep of the diffusion process and the class label. Similarly, we use embeddings of the timestep t and c_{vec} as inputs to the modulation mechanism. However, as the pooled text representation retains only coarse-grained information about the text input (Podell 等, 2023), the network also requires information from the sequence representation c_{txt} .

我们的架构基于 DiT (Peebles & Xie, 2023)。DiT 仅考虑类条件图像生成, 使用调制机制将网络条件化为扩散过程的时间步和类标签。类似地, 我们使用时间步 t 和 c_{vec} 的嵌入作为调制机制的输入。但是, 由于池化的文本表示仅保留关于文本输入的粗粒度信息 (Podell 等, 2023), 网络也需要来自序列表示 c_{txt} 的信息。

We construct a sequence consisting of embeddings of the text and image inputs. Specifically, we add positional encodings and flatten 2×2 patches of the latent pixel representation $x \in \mathbb{R}^{h \times w \times c}$ to a patch encoding sequence of length $\frac{1}{2} \cdot h \cdot \frac{1}{2} \cdot w$. After embedding this patch encoding and the text encoding c_{txt} to a common dimensionality, we concatenate the two sequences. We then follow DiT and apply a sequence of modulated attention and MLPs.

我们构建由文本和图像输入的嵌入组成的序列。

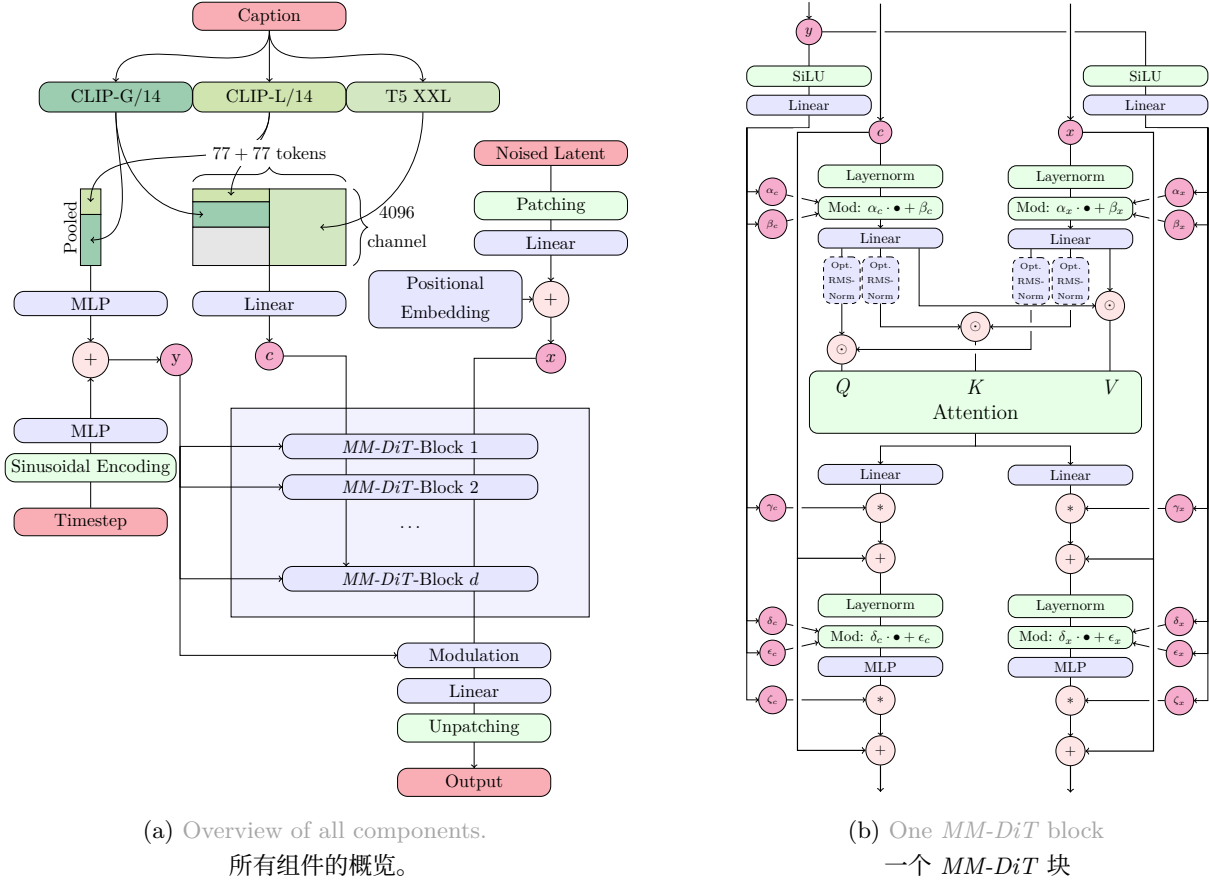


图 2. **Our model architecture.** Concatenation is indicated by $\odot$ and element-wise multiplication by $*$. The RMS-Norm 我们的模型架构。连接用 $\odot$ 表示，元素乘法用 $*$ 表示。可以添加 Q 和 K 的 RMS-Norm 来稳定训练运行。最好放大后查看。

具体地，我们添加位置编码并将潜在像素表示 $x \in \mathbb{R}^{h \times w \times c}$ 的 2×2 patch 展平为长度为 $\frac{1}{2} \cdot h \cdot \frac{1}{2} \cdot w$ 的 patch 编码序列。将这个 patch 编码和文本编码 c_{txt} 嵌入到通用维度后，我们连接两个序列。然后我们遵循 DiT，应用一系列调制注意力和 MLP。

Since text and image embeddings are conceptually quite different, we use two separate sets of weights for the two modalities. As shown in Figure 2b, this is equivalent to having two independent transformers for each modality, but joining the sequences of the two modalities for the attention operation, such that both representations can work in their own space yet take the other one into account.

由于文本和图像嵌入在概念上差异很大，我们为两种模态使用两个独立的权重集。如 Figure 2b 所示，这等价于为每种模态有两个独立的 Transformer，但在注意力操作中连接两种模态的序列，使得两种表示可以在各自的空间中工作，同时考虑另一个。

For our scaling experiments, we parameterize the size of the model in terms of the model’s depth d , i.e. the number of attention blocks, by setting the hidden size to $64 \cdot d$ (expanded to $4 \cdot 64 \cdot d$ channels in the MLP blocks), and the number of attention heads equal to d .

对于我们的缩放实验，我们用模型的深度 d （即注意力块的数量）参数化模型的大小，通过将隐藏大小设置为 $64 \cdot d$ （在 MLP 块中扩展到 $4 \cdot 64 \cdot d$ 通道），并将注意力头的数量设置为 d 。

5. 实验

5.1. 改进修正流

We aim to understand which of the approaches for simulation-free training of normalizing flows as in Equation 1 is the most efficient. To enable comparisons across different approaches, we control for the optimization algorithm, the model architecture, the dataset and sam-

variant	rank averaged over		
	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

表 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. **变体的全局排名。**对于该排名，我们应用非支配排序，对 EMA 和非 EMA 权重、两个数据集和不同的采样设置进行平均。

plers. In addition, the losses of different approaches are incomparable and also do not necessarily correlate with the quality of output samples; hence we need evaluation metrics that allow for a comparison between approaches. We train models on ImageNet (Russakovsky 等, 2014) and CC12M (Changpinyo 等, 2021), and evaluate both the training and the EMA weights of the models during training using validation losses, CLIP scores (Radford 等, 2021; Hessel 等, 2021), and FID (Heusel 等, 2017) under different sampler settings (different guidance scales and sampling steps). We calculate the FID on CLIP features as proposed by (Sauer 等, 2021). All metrics are evaluated on the COCO-2014 validation split (Lin 等, 2014). Full details on the training and sampling hyperparameters are provided in Appendix B.3.

我们的目标是理解 Equation 1 中免仿真训练归一化流的哪种方法最高效。为了能在不同方法间进行比较，我们控制优化算法、模型架构、数据集和采样器。此外，不同方法的损失函数不可直接比较，也不一定与输出样本质量相关。因此我们需要能在方法间进行比较的评价指标。

variant	ImageNet		CC12M	
	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	<i>90.34</i>
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	<u>0.233</u>	120.84
rf/mode(1.75)	<i>0.253</i>	44.39	0.218	94.06
rf/lognorm(1.00, 0.60)	<u>0.254</u>	114.26	0.234	147.69
rf/lognorm(-0.50, 1.00)	0.248	<u>45.64</u>	0.219	89.70
rf/lognorm(0.00, 1.00)	0.250	<i>45.78</i>	<i>0.224</i>	<u>89.91</u>

表 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. **不同变体的指标。**采用 25 个采样步的不同变体的 FID 和 CLIP 分数。我们突出显示了 **最好**、第二好 和 *第三好* 的条目。

我们在 ImageNet (Russakovsky 等, 2014) 和 CC12M (Changpinyo 等, 2021) 上训练模型，并在训练过程中使用验证损失、CLIP 分数 (Radford 等, 2021; Hessel 等, 2021) 和 FID (Heusel 等, 2017) 评估训练权重和 EMA 权重，同时考虑不同的采样器设置（不同的引导强度和采样步数）。我们按 (Sauer 等, 2021) 的方法计算 CLIP 特征上的 FID。所有指标都在 COCO-2014 验证集 (Lin 等, 2014) 上进行评估。训练和采样超参数的详细信息见 Appendix B.3。

5.1.1. 结果

我们在两个数据集上训练每种的 61 种不同的公式。我们包含来自 Section 3 的以下变体：

- ϵ - 和 v 预测损失，采用线性 (eps/linear、v/linear) 和余弦 (eps/cos、v/cos) 调度。
- RF 损失搭配 $\pi_{\text{mode}}(t; s)$ (rf/mode(s))，其中 s 的 7 个值在 -1 到 1.75 之间均匀分布，以及 $s = 1.0$ 和 $s = 0$ 的情况，后者对应均匀时间步采样 (rf/mode)。
- RF 损失搭配 $\pi_{\text{ln}}(t; m, s)$ (rf/lognorm(m, s))，网格中有 30 个 (m, s) 值，其中 m 在 -1 到 1 之间均匀分布， s 在 0.2 到 2.2 之间均匀分布。
- RF 损失搭配 $\pi_{\text{CosMap}}(t)$ (rf/cosmap)。

- EDM ($\text{edm}(P_m, P_s)$), P_m 的 15 个值在 -1.2 到 1.2 之间均匀分布, P_s 在 0.6 到 1.8 之间均匀分布。注意 $P_m, P_s = (-1.2, 1.2)$ 对应 (Karras 等, 2022) 中的参数。
- EDM 搭配使其匹配 rf 的 log-SNR 权重的调度 (edm/rf) 和匹配 v/cos 的 log-SNR 权重的调度 (edm/cos)。

For each run, we select the step with minimal validation loss when evaluated with EMA weights and then collect CLIP scores and FID obtained with 6 different sampler settings both with and without EMA weights.

对于每次运行, 我们选择用 EMA 权重评估时验证损失最小的步骤, 然后收集用 6 种不同采样器设置 (有和没有 EMA 权重) 获得的 CLIP 分数和 FID。

For all 24 combinations of sampler settings, EMA weights, and dataset choice, we rank the different formulations using a non-dominated sorting algorithm. For this, we repeatedly compute the variants that are Pareto optimal according to CLIP and FID scores, assign those variants the current iteration index, remove those variants, and continue with the remaining ones until all variants get ranked. Finally, we average those ranks over the 24 different control settings.

对于采样器设置、EMA 权重和数据集选择的所有 24 种组合, 我们使用非主导排序算法对不同公式进行排名。方法如下: 反复识别在 CLIP 和 FID 分数上达到 Pareto 最优的变体, 为其分配当前迭代索引, 删除该变体, 继续处理剩余变体, 直至全部排名完成。最后在 24 种控制设置上平均这些排名。

We present the results in Tab. 1, where we only show the two best-performing variants for those variants that were evaluated with different hyperparameters. We also show ranks where we restrict the averaging over sampler settings with 5 steps and with 50 steps.

我们在表 1 中呈现结果, 其中对于用不同超参数评估的变体, 我们只展示表现最好的两个变体。我们还显示了限制在 5 步和 50 步采样器设置上的平均排名。

We observe that $\text{rf}/\text{lognorm}(0.00, 1.00)$ consistently achieves a good rank. It outperforms a rectified flow formulation with uniform timestep sampling (rf) and

thus confirms our hypothesis that intermediate timesteps are more important. Among all the variants, *only* rectified flow formulations with modified timestep sampling perform better than the LDM-Linear (Rombach 等, 2022) formulation (eps/linear) used previously.

我们观察到 $\text{rf}/\text{lognorm}(0.00, 1.00)$ 始终达到较好的排名。它优于具有均匀时间步采样的修正流公式 (rf), 从而验证了中间时间步更重要的假设。在所有变体中, 仅有带有修改时间步采样的修正流公式表现优于之前使用的 LDM-Linear (Rombach 等, 2022) 公式 (eps/linear)。

We also observe that some variants perform well in some settings but worse in others, *e.g.* $\text{rf}/\text{lognorm}(0.50, 0.60)$ is the best-performing variant with 50 sampling steps but much worse (average rank 8.5) with 5 sampling steps. We observe a similar behavior with respect to the two metrics in Tab. 2. The first group shows representative variants and their metrics on both datasets with 25 sampling steps. The next group shows the variants that achieve the best CLIP and FID scores. With the exception of $\text{rf}/\text{mode}(1.75)$, these variants typically perform very well in one metric but relatively badly in the other. In contrast, we once again observe that $\text{rf}/\text{lognorm}(0.00, 1.00)$ achieves good performance across metrics and datasets, where it obtains the third-best scores two out of four times and once the second-best performance.

我们也观察到某些变体在某些设置下表现良好, 而在其他设置下表现较差。例如, $\text{rf}/\text{lognorm}(0.50, 0.60)$ 在 50 个采样步数下是表现最好的变体, 但在 5 个采样步数下表现要差得多 (平均排名 8.5)。在表 2 的两个指标中我们观察到类似的行为。第一组显示代表性变体及其在两个数据集上采用 25 个采样步数的指标。下一组显示达到最佳 CLIP 和 FID 分数的变体。除 $\text{rf}/\text{mode}(1.75)$ 外, 这些变体通常在一个指标上表现非常好, 但在另一个指标上表现相对较差。相比之下, 我们再次观察到 $\text{rf}/\text{lognorm}(0.00, 1.00)$ 在各指标和数据集上都取得良好表现, 其中四次中有两次获得第三最佳分数, 一次获得第二最佳表现。

Finally, we illustrate the qualitative behavior of different formulations in Figure 3, where we use different colors for different groups of formulations (edm , rf , eps and v). Rectified flow formulations generally perform well

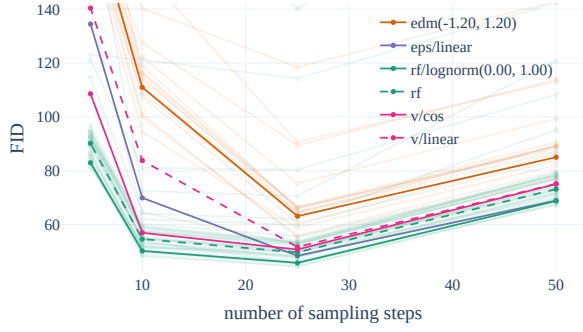


图 3. Rectified flows are sample efficient. Rectified Flows perform better than other formulations when sampling fewer steps. For 25 and more steps, only rf/lognorm(0.00, 1.00) remains competitive to 修正流具有采样效率。修正流在采样更少步数时比其他公式表现更好。对于 25 步及以上，只有 rf/lognorm(0.00, 1.00) 对 eps/linear 保持竞争力。

and, compared to other formulations, their performance degrades less when reducing the number of sampling steps.

最后，我们在 Figure 3 中说明了不同公式的定性行为，其中我们对不同的公式组使用不同的颜色 (edm、rf、eps 和 v)。修正流公式总体上表现良好，与其他公式相比，减少采样步数时其性能下降幅度较小。

5.2. 改进模态特定表示

Having found a formulation in the previous section that allows rectified flow models to not only compete with established diffusion formulations such as LDM-Linear (Rombach 等, 2022) or EDM (Karras 等, 2022), but even outperforms them, we now turn to the application of our formulation to high-resolution text-to-image synthesis. Accordingly, the final performance of our algorithm depends not only on the training formulation, but also on the parameterization via a neural network and the quality of the image and text representations we use. In the following sections, we describe how we improve all these components before scaling our final method in Section 5.3.

在前面的章节中，我们找到了一个公式，使修正流模型不仅能与 LDM-Linear (Rombach 等, 2022) 或 EDM (Karras 等, 2022) 等成熟的扩散公式竞争，甚至还能超越它们。现在我们将该公式应用于高分辨率文本到图像合成。相应地，我们算法的最终性能不仅取决于训练公式，还取决于通过神经网络的参数化以及我们使用的图像和文本表示的质量。在以下章节中，我们描述在 Section 5.3 中缩放最终方法之前如何改进

指标	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

表 3. Improved Autoencoders. Reconstruction performance metrics for different channel configurations. The 改进的自编码器在不同通道配置下的重建性能指标。所有模型的下采样因子均为 $f = 8$ 。

这些所有组件。

5.2.1. 改进自编码器

Latent diffusion models achieve high efficiency by operating in the latent space of a pretrained autoencoder (Rombach 等, 2022), which maps an input RGB $X \in \mathbb{R}^{H \times W \times 3}$ into a lower-dimensional space $x = E(X) \in \mathbb{R}^{h \times w \times d}$. The reconstruction quality of this autoencoder provides an upper bound on the achievable image quality after latent diffusion training. Similar to Dai 等 (2023), we find that increasing the number of latent channels d significantly boosts reconstruction performance, see Table 3. Intuitively, predicting latents with higher d is a more difficult task, and thus models with increased capacity should be able to perform better for larger d , ultimately achieving higher image quality. We confirm this hypothesis in Figure 10, where we see that the $d = 16$ autoencoder exhibits better scaling performance in terms of sample FID. For the remainder of this paper, we thus choose $d = 16$.

潜在扩散模型通过在预训练自编码器的潜在空间中运行来实现高效率 (Rombach 等, 2022)，该自编码器将输入 RGB 图像 $X \in \mathbb{R}^{H \times W \times 3}$ 映射到低维空间 $x = E(X) \in \mathbb{R}^{h \times w \times d}$ 。该自编码器的重建质量为潜在扩散训练后可达到的图像质量提供上界。与 Dai 等 (2023) 类似，我们发现增加潜在通道数 d 能显著提升重建性能 (见 Table 3)。增加 d 使潜在表示预测更具挑战性，因此容量更大的模型能在更高的 d 下表现更优，最终获得更高的图像质量。我们在 Figure 10 中验证了这一假设， $d = 16$ 自编码器在样本 FID 上展现出更佳的缩放特性。基于这些结果，我们在后续实验中采用 $d = 16$ 。

5.2.2. 改进描述文本

Betker 等 (2023) demonstrated that synthetically gen-



a space elevator, cinematic scifi art



A cheeseburger with juicy beef patties and melted cheese sits on top of a toilet that looks like a throne and stands in the middle of the royal chamber.



a hole in the floor of my bathroom with small gremlins living in it



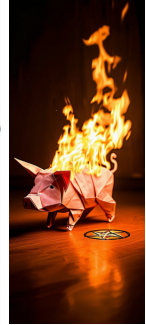
a small office made out of car parts



This dreamlike digital art captures a vibrant, kaleidoscopic bird in a lush rainforest.



human life depicted entirely out of fractals



an origami pig on fire in the middle of a dark room with a pentagram on the floor



an old rusted robot wearing pants and a jacket riding skis in a supermarket.



smiling cartoon dog sits at a table, coffee mug on hand, as a room goes up in flames. "This is fine," the dog assures himself.



A whimsical and creative image depicting a hybrid creature that is a mix of a waffle and a hippopotamus. This imaginative creature features the distinctive, bulky body of a hippo, but with a texture and appearance resembling a golden-brown, crispy waffle. The creature might have elements like waffle squares across its skin and a syrup-like sheen. It's set in a surreal environment that playfully combines a natural water habitat of a hippo with elements of a breakfast table setting, possibly including oversized utensils or plates in the background. The image should evoke a sense of playful absurdity and culinary fantasy.

erated captions can greatly improve text-to-image models trained at scale. This is due to the oftentimes simplistic nature of the human-generated captions that come with large-scale image datasets, which overly focus on the image subject and usually omit details describing the background or composition of the scene, or, if applicable, displayed text (Betker 等, 2023). We follow their approach and use an off-the-shelf, state-of-the-art vision-language model, *CogVLM* (Wang 等, 2023), to create synthetic annotations for our large-scale image dataset. As synthetic captions may cause a text-to-image model to forget about certain concepts not present in the VLM’s knowledge corpus, we use a ratio of 50 % original and 50 % synthetic captions.

Betker 等 (2023) 证明合成描述文本能显著改进大规模文本到图像模型。这是因为大规模图像数据集中的人工描述文本通常过于简化, 过度关注图像主题, 常常遗漏背景/场景构成的细节和文字内容 (Betker 等, 2023)。我们沿用此方法, 使用最先进的视觉语言模型 *CogVLM* (Wang 等, 2023) 为大规模数据集生成合成描述文本。为防止合成文本导致模型遗忘 VLM 知识库外的概念, 我们混用 50% 原始和 50% 合成描述文本。

To assess the effect of training on this caption mix, we train two $d = 15$ *MM-DiT* models for 250k steps, one on only original captions and the other on the 50/50 mix. We evaluate the trained models using the GenEval benchmark (Ghosh 等, 2023) in Table 4. The results demonstrate that the model trained with the addition of synthetic captions clearly outperforms the model that only utilizes original captions. We thus use the 50/50 synthetic/original caption mix for the remainder of this work.

为了评估用此描述文本混合进行训练的效果, 我们训练了两个 $d = 15$ *MM-DiT* 模型共 250k 步, 一个仅用原始描述文本, 另一个用 50/50 混合。我们在 Table 4 中使用 GenEval 基准 (Ghosh 等, 2023) 评估训练后的模型。结果表明使用合成描述文本进行训练的模型明显优于仅使用原始描述文本的模型。因此, 在本工作的其余部分我们采用 50/50 合成/原始描述文本混合。

5.2.3. 改进文本到图像骨干网络

	原始描述文本	50/50 混合
	成功率 [%]	成功率 [%]
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

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

改进的描述文本。使用合成 (通过 CogVLM (Wang 等, 2023)) 和原始描述文本的 50/50 混合比例改进了文本到图像性能。通过 GenEval (Ghosh 等, 2023) 基准进行评估。

In this section, we compare the performance of existing transformer-based diffusion backbones with our novel multimodal transformer-based diffusion backbone, *MM-DiT*, as introduced in Section 4. *MM-DiT* is specifically designed to handle different domains, here text and image tokens, using (two) different sets of trainable model weights. More specifically, we follow the experimental setup from Section 5.1 and compare text-to-image performance on CC12M of DiT, CrossDiT (DiT but with cross-attending to the text tokens instead of sequence-wise concatenation (Chen 等, 2023)) and our *MM-DiT*. For *MM-DiT*, we compare models with two sets of weights and three sets of weights, where the latter handles the CLIP (Radford 等, 2021) and T5 (Raffel 等, 2019) tokens (*c.f.* Section 4) separately. Note that DiT (w/ concatenation of text and image tokens as in Section 4) can be interpreted as a special case of *MM-DiT* with one shared set of weights for all modalities. Finally, we consider the UViT (Hoogeboom 等, 2023) architecture as a hybrid between the widely used UNets and transformer variants.

本节比较现有 Transformer 扩散骨干与我们提出的多模态 Transformer 扩散骨干 *MM-DiT* (见 Section 4) 的性能。*MM-DiT* 为多模态设计, 采用分离的参数集处理不同的领域 (文本和图像 token)。

具体地, 我们采用 Section 5.1 的实验设置, 在 CC12M 上比较 DiT、CrossDiT (交叉注意文本 token 而非拼接 (Chen 等, 2023)) 和我们的 *MM-DiT*。对

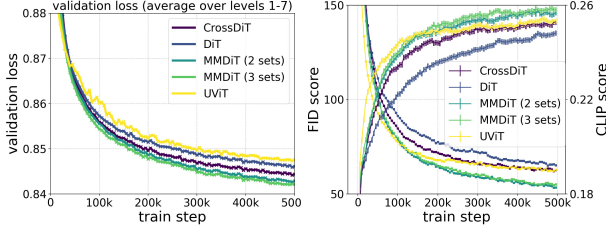


图 4. Training dynamics of model architectures. Comparative analysis of *DiT*, *CrossDiT*, *UViT*, and *MM-DiT* on CC12M, focusing on validation loss, CLIP score, and FID. Our proposed *MM-DiT* performs favorably across all metrics.

模型架构的训练动态。对 *DiT*、*CrossDiT*、*UViT* 和 *MM-DiT* 在 CC12M 上的比较分析，重点关注验证损失、CLIP 分数和 FID。我们提出的 *MM-DiT* 在所有指标上表现良好。

MM-DiT，我们测试两组权重（共享全部 token）和三组权重（分别处理 CLIP (Radford 等, 2021)、T5 (Raffel 等, 2019) 和图像 token，见 Section 4) 两个版本。注意标准 *DiT* 可视为 *MM-DiT* 的特例（所有模式共享一组权重）。此外，我们还评估 *UViT* (Hoogeboom 等, 2023)——一个介于 UNet 和 Transformer 间的混合架构。

We analyze the convergence behavior of these architectures in Figure 4: Vanilla *DiT* underperforms *UViT*. The cross-attention *DiT* variant *CrossDiT* achieves better performance than *UViT*, although *UViT* seems to learn much faster initially. Our *MM-DiT* variant significantly outperforms the cross-attention and vanilla variants. We observe only a small gain when using three parameter sets instead of two (at the cost of increased parameter count and VRAM usage), and thus opt for the former option for the remainder of this work.

我们在 Figure 4 中分析这些架构的收敛行为：普通 *DiT* 表现不如 *UViT*。交叉注意 *DiT* 变体 *CrossDiT* 表现优于 *UViT*，尽管 *UViT* 初期学习速度更快。我们的 *MM-DiT* 变体显著优于交叉注意和普通变体。我们观察到使用三组参数集而非两组只能获得很小的改进（但代价是参数数量和 VRAM 使用量增加），因此在本工作的其余部分采用前者。

5.3. 大规模训练

Before scaling up, we filter and preencode our data to ensure safe and efficient pretraining. Then, all previous considerations of diffusion formulations, architectures,

and data culminate in the last section, where we scale our models up to 8B parameters.

大规模训练前，我们先进行数据过滤与预编码以保证安全高效的预训练。随后，我们将前面关于扩散公式、架构和数据处理的所有经验整合应用于模型缩放，最终训练至 8B 参数。

5.3.1. 数据预处理

预训练风险缓解 训练数据对生成模型的能力有重大影响。因此，数据过滤能有效地约束不良能力 (Nichol, 2022)。在大规模训练前，我们对以下类别进行数据过滤：(i) 性内容：我们使用 NSFW 检测模型过滤露骨内容。(ii) 美学：我们删除我们的评分系统预测分数较低的图像。(iii) 内容重复：我们使用基于聚类的去重方法从训练数据中删除感知和语义重复；见 Appendix E.2。

Precomputing Image and Text Embeddings

Our model uses the output of multiple pretrained, frozen networks as inputs (autoencoder latents and text encoder representations). Since these outputs are constant during training, we precompute them once for the entire dataset. We provide a detailed discussion of our approach in Appendix E.1.

预计算图像和文本嵌入 我们的模型使用多个预训练的冻结网络的输出作为输入（自编码器潜在表示和文本编码器表示）。由于这些输出在训练过程中是常数，我们为整个数据集预计算一次。我们在 Appendix E.1 中提供了对我们方法的详细讨论。

5.3.2. 高分辨率微调

QK-Normalization In general, we pretrain all of our models on low-resolution images of size 256^2 pixels. Next, we finetune our models on higher resolutions with mixed aspect ratios (see next paragraph for details). We find that, when moving to high resolutions, mixed precision training can become unstable and the loss diverges. This can be remedied by switching to full precision training — but comes with a $\sim 2\times$ performance drop compared to mixed-precision training. A more efficient alternative is reported in the (discriminative) ViT literature: Dehghani 等 (2023) observe that the training of large vision transformer models diverges because the attention entropy grows uncontrollably. To avoid this, Dehghani 等 (2023)

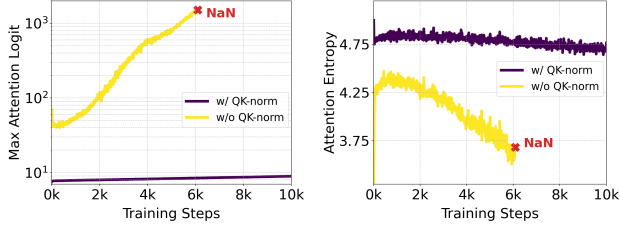


图 5. **Effects of QK-normalization.** Normalizing the Q- and K-embeddings before calculating the attention matrix prevents the attention-logit growth instability (*left*), which causes the attention entropy to collapse (*right*), and has been previously reported in the discriminative ViT literature (Dehghani 等, 2023; Wortsman 等, 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.

QK 归一化的效果。在计算注意力矩阵前对 Q 和 K 嵌入进行归一化可以防止注意力 logit 增长不稳定性 (左), 该不稳定性会导致注意力熵崩溃 (右) 并且已在判别性 ViT 文献中报道过 (Dehghani 等, 2023; Wortsman 等, 2023)。与这些以前的工作相反, 我们在网络的最后 Transformer 块中观察到这种不稳定性。最大注意力 logit 和注意力熵显示为 2B ($d=24$) 模型最后 5 个块的平均值。

propose to normalize Q and K before the attention operation. We follow this approach and use RMSNorm (Zhang & Sennrich, 2019) with learnable scale in both streams of our MMDiT architecture for our models, see Figure 2. As demonstrated in Figure 5, the additional normalization prevents the attention logit growth instability, confirming findings by Dehghani 等 (2023) and Wortsman 等 (2023) and enables efficient training at bf16-mixed (Chen 等, 2019) precision when combined with $\epsilon = 10^{-15}$ in the AdamW (Loshchilov & Hutter, 2017) optimizer. This technique can also be applied on pretrained models that have not used qk-normalization during pretraining: The model quickly adapts to the additional normalization layers and trains more stably. Finally, we would like to point out that although this method can generally help to stabilize the training of large models, it is not a universal recipe and may need to be adapted depending on the exact training setup.

QK 归一化 通常, 我们在大小为 256^2 像素的低分辨率图像上预训练所有模型。接下来, 我们在具有混合宽高比的高分辨率上微调模型 (详见下一段)。我们发现, 当移至高分辨率时, 混合精度训练可能变得不稳定, 损失发散。这可以通过切换到全精度训练来解决——但相比混合精度训练会导致约 $\sim 2\times$ 的性能下降。在 (判别

性) ViT 文献中报告了更高效的替代方案: Dehghani 等 (2023) 观察到大型视觉 Transformer 模型的训练发散, 因为注意力熵不可控地增长。为避免这种情况, Dehghani 等 (2023) 提议在注意力操作前对 Q 和 K 进行归一化。我们采用此方法, 在 MMDiT 架构的两个流中对模型使用具有可学习缩放的 RMSNorm (Zhang & Sennrich, 2019) (见 Figure 2)。如 Figure 5 所示, 额外的归一化防止注意力 logit 增长不稳定性, 验证了 Dehghani 等 (2023) 和 Wortsman 等 (2023) 的发现, 并在与 AdamW (Loshchilov & Hutter, 2017) 优化器中的 $\epsilon = 10^{-15}$ 结合时能够实现 bf16-混合 (Chen 等, 2019) 精度的高效训练。该技术也可应用于在预训练期间未使用 qk 归一化的预训练模型: 模型快速适应额外的归一化层并更稳定地训练。最后, 我们想指出, 尽管此方法通常能帮助稳定大型模型的训练, 但它不是通用的方案, 可能需要根据具体的训练设置进行调整。

Positional Encodings for Varying Aspect Ratios

不同宽高比的位置编码 After training on a fixed 256×256 resolution we aim to (i) increase the resolution and resolution and (ii) enable inference with flexible aspect ratios. Since we use 2d positional frequency embeddings we have to adapt them based on the resolution. In the multi-aspect ratio setting, a direct interpolation of the embeddings as in (Dosovitskiy 等, 2020) would not reflect the side lengths correctly. Instead we use a combination of extended and interpolated position grids which are subsequently frequency embedded.

在固定 256×256 分辨率上训练后, 我们的目标是 (i) 提高分辨率, (ii) 启用灵活宽高比的推理。由于我们使用 2D 位置频率嵌入, 必须根据分辨率调整它们。在多宽高比设置中, 如 (Dosovitskiy 等, 2020) 中的直接嵌入插值不会正确反映边长。相反, 我们使用扩展和插值位置网格的组合, 然后进行频率嵌入。

For a target resolution of S^2 pixels, we use bucketed sampling (NovelAI, 2022; Podell 等, 2023) such that each batch consists of images of a homogeneous size $H \times W$, where $H \cdot W \approx S^2$. For the maximum and minimum training aspect ratios, this results in the maximum values for width, $W_{\max}$, and height, $H_{\max}$, that will be encountered. Let $h_{\max} = H_{\max}/16, w_{\max} = W_{\max}/16$ and

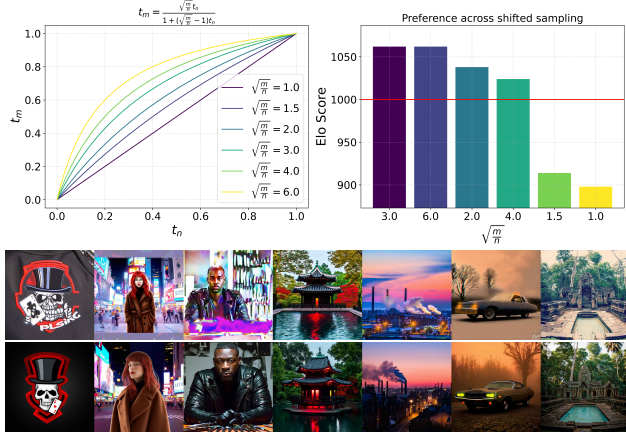


图 6. Timestep shifting at higher resolutions. *Top right*: Human quality preference rating when applying the shifting based on Equation (22). *Bottom row*: A 512^2 model trained and sampled with $\sqrt{m/n} = 1.0$ (*top*) and $\sqrt{m/n} = 3.0$ (*bottom*). See Section 5.3.2.

高分辨率下的时间步移位。右上: 应用基于 Equation (22) 的移位时的人类质量偏好评分。底行: 用 $\sqrt{m/n} = 1.0$ (上) 和 $\sqrt{m/n} = 3.0$ (下) 训练和采样的 512^2 模型。见 Section 5.3.2。

$s = S/16$ be the corresponding sizes in latent space (a factor 8) after patching (a factor 2). Based on these values, we construct a vertical position grid with the values $((p - \frac{h_{\max}-s}{2}) \cdot \frac{256}{S})_{p=0}^{h_{\max}-1}$ and correspondingly for the horizontal positions. We then center-crop from the resulting positional 2d grid before embedding it.

对于目标分辨率 S^2 像素, 我们使用桶式采样 (NovelAI, 2022; Podell 等, 2023), 使得每个批次包含同质尺寸 $H \times W$ 的图像, 其中 $H \cdot W \approx S^2$. 对于最大和最小训练宽高比, 这产生了将遇到的宽度最大值 $W_{\max}$ 和高度最大值 $H_{\max}$. 设 $h_{\max} = H_{\max}/16$, $w_{\max} = W_{\max}/16$ 和 $s = S/16$ 为潜在空间 (因子 8) 中的对应尺寸, 在切分 (因子 2) 之后。基于这些值, 我们构建一个垂直位置网格, 具有值 $((p - \frac{h_{\max}-s}{2}) \cdot \frac{256}{S})_{p=0}^{h_{\max}-1}$, 水平位置相应类似。然后我们从生成的位置 2D 网格中心裁剪后再进行嵌入。

Resolution-dependent shifting of timestep schedules Intuitively, since higher resolutions have more pixels, we need more noise to destroy their signal. Assume we are working in a resolution with $n = H \cdot W$ pixels. Now, consider a "constant" image, i.e. one where every pixel has the value c . The forward process produces $z_t = (1-t)c\mathbb{1} + t\epsilon$, where both $\mathbb{1}$ and $\epsilon \in \mathbb{R}^n$. Thus, z_t provides n observations of the random variable $Y = (1-t)c + t\eta$

with c and η in $\mathbb{R}$, and η follows a standard normal distribution. Thus, $\mathbb{E}(Y) = (1-t)c$ and $\sigma(Y) = t$. We can therefore recover c via $c = \frac{1}{1-t}\mathbb{E}(Y)$, and the error between c and its sample estimate $\hat{c} = \frac{1}{1-t} \sum_{i=1}^n z_{t,i}$ has a standard deviation of $\sigma(t, n) = \frac{t}{1-t} \sqrt{\frac{1}{n}}$ (because the standard error of the mean for Y has deviation $\frac{t}{\sqrt{n}}$). So if one already knows that the image z_0 was constant across its pixels, $\sigma(t, n)$ represents the degree of uncertainty about z_0 . For example, we immediately see that doubling the width and height leads to half the uncertainty at any given time $0 < t < 1$. But, we can now map a timestep t_n at resolution n to a timestep t_m at resolution m that results in the same degree of uncertainty via the ansatz $\sigma(t_n, n) = \sigma(t_m, m)$. Solving for t_m gives

分辨率依赖的时间步调度移位 从直觉上讲, 由于更高的分辨率有更多像素, 我们需要更多噪声来破坏它们的信号。假设我们在具有 $n = H \cdot W$ 像素的分辨率下工作。现在考虑一个“常数”图像, 即每个像素都有值 c 的图像。前向过程产生 $z_t = (1-t)c\mathbb{1} + t\epsilon$, 其中 $\mathbb{1}$ 和 $\epsilon \in \mathbb{R}^n$ 。因此 z_t 提供了随机变量 $Y = (1-t)c + t\eta$ 的 n 个观测, 其中 c 和 $\eta \in \mathbb{R}$, 且 η 服从标准正态分布。因此 $\mathbb{E}(Y) = (1-t)c$ 且 $\sigma(Y) = t$ 。我们可以通过 $c = \frac{1}{1-t}\mathbb{E}(Y)$ 恢复 c , c 与其样本估计 $\hat{c} = \frac{1}{1-t} \sum_{i=1}^n z_{t,i}$ 之间的误差标准差为 $\sigma(t, n) = \frac{t}{1-t} \sqrt{\frac{1}{n}}$ (因为 Y 的平均值标准误差为 $\frac{t}{\sqrt{n}}$)。因此如果已经知道图像 z_0 在其像素间是常数, $\sigma(t, n)$ 代表关于 z_0 的不确定性程度。例如, 我们立即看到宽度和高度加倍会导致任何给定时刻 $0 < t < 1$ 的不确定性减半。但是, 我们现在可以将分辨率 n 处的时间步 t_n 映射到分辨率 m 处的时间步 t_m , 通过引理 $\sigma(t_n, n) = \sigma(t_m, m)$ 得到相同的不确定性程度。求解 t_m 得到

$$t_m = \frac{\sqrt{\frac{m}{n}} t_n}{1 + (\sqrt{\frac{m}{n}} - 1) t_n} \quad (22)$$

We visualize this shifting function in Figure 6. Note that the assumption of constant images is not realistic. To find good values for the shift value $\alpha := \sqrt{\frac{m}{n}}$ during inference, we apply them to the sampling steps of a model trained at resolution 1024×1024 and run a human preference study. The results in Figure 6 show a strong preference for samples with shifts greater than 1.5 but less drastic differences among the higher shift values. In our subsequent experiments, we thus use a shift value of $\alpha = 3.0$ both during training and sampling at resolution 1024×1024 . A quali-

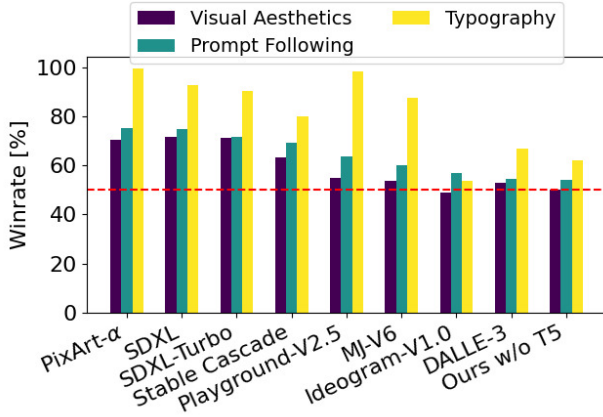


图 7. Human Preference Evaluation against current closed and open SOTA generative image models. Our 8B model compares favorably against current state-of-the-art text-to-image models when evaluated on the parti-prompts (Yu 等, 2022) across the categories *visual quality*, *prompt following* and *typography generation*. 针对当前开源和闭源 SOTA 生成图像模型的人类偏好评估。我们的 8B 模型与当前最先进的文本到图像模型进行比较，在 parti-prompts (Yu 等, 2022) 上进行评估，涵盖视觉质量、提示遵循和文字排版生成等类别。

tative comparison between samples after 8k training steps with and without such a shift can be found in Figure 6. Finally, note that Equation 22 implies a log-SNR shift of $\log \frac{n}{m}$ similar to (Hooeboom 等, 2023):

我们在 Figure 6 中可视化这个移位函数。注意常数图像的假设不切实际。为了在推理时找到移位值 $\alpha := \sqrt{\frac{m}{n}}$ 的良好值，我们将其应用于在分辨率 1024×1024 处训练的模型的采样步数，并进行人类偏好研究。Figure 6 中的结果显示对移位值大于 1.5 的样本有很强的偏好，但较高移位值之间差异不那么明显。在我们的后续实验中，我们在分辨率 1024×1024 处的训练和采样中都使用移位值 $\alpha = 3.0$ 。8k 个训练步数后有和没有这种移位的样本的定性比较见 Figure 6。最后，注意 Equation 22 蕴含类似于 (Hooeboom 等, 2023) 的 log-SNR 移位 $\log \frac{n}{m}$ ：

$$\lambda_{tm} = 2 \log \frac{1 - t_n}{\sqrt{\frac{m}{n} t_n}} \quad (23)$$

$$= \lambda_{t_n} - 2 \log \alpha = \lambda_{t_n} - \log \frac{m}{n}. \quad (24)$$

After the shifted training at resolution 1024×1024 , we align the model using Direct Preference Optimization (DPO) as described in Appendix C.

在分辨率 1024×1024 的移位训练之后，我们使用 Appendix C 中描述的直接偏好优化 (DPO) 来对齐模型。

5.3.3. 结果

In Figure 8, we examine the effect of training our *MM-DiT* at scale. For images, we conduct a large scaling study and train models with different numbers of parameters for 500k steps on 256^2 pixels resolution using preencoded data, *c.f.* Appendix E.1, with a batch size of 4096. We train on 2×2 patches (Peebles & Xie, 2023), and report validation losses on the CoCo dataset (Lin 等, 2014) every 50k steps. In particular, to reduce noise in the validation loss signal, we sample loss levels equidistant in $t \in (0, 1)$ and compute validation loss for each level separately. We then average the loss across all but the last ($t = 1$) levels.

在 Figure 8 中，我们检验大规模训练我们的 *MM-DiT* 的效果。对于图像，我们进行大规模缩放研究，在 256^2 像素分辨率上用预编码数据（见 Appendix E.1）训练不同参数数量的模型共 500k 步，批大小为 4096。我们在 2×2 patch 上训练 (Peebles & Xie, 2023)，每 50k 步在 CoCo 数据集 (Lin 等, 2014) 上报告验证损失。特别地，为了减少验证损失信号中的噪声，我们采样在 $t \in (0, 1)$ 中等距的损失水平，并分别为每个水平计算验证损失。然后我们在除最后一个 ($t = 1$) 之外的所有水平上平均损失。

Similarly, we conduct a preliminary scaling study of our *MM-DiT* on videos. To this end we start from the pre-trained image weights and additionally use a 2x temporal patching. We follow Blattmann 等 (2023b) and feed data to the pretrained model by collapsing the temporal into the batch axis. In each attention layer we rearrange the representation in the visual stream and add a full attention over all spatio-temporal tokens after the spatial attention operation before the final feedforward layer. Our video models are trained for 140k steps with a batch size of 512 on videos comprising 16 frames with 256^2 pixels. We report validation losses on the Kinetics dataset (Carreira & Zisserman, 2018) every 5k steps. Note that our reported FLOPs for video training in Figure 8 are only FLOPs from video training and do not include the FLOPs from image pretraining.

类似地，我们对视频进行了 *MM-DiT* 的初步缩放研究。为此我们从预训练的图像权重开始，另外使用

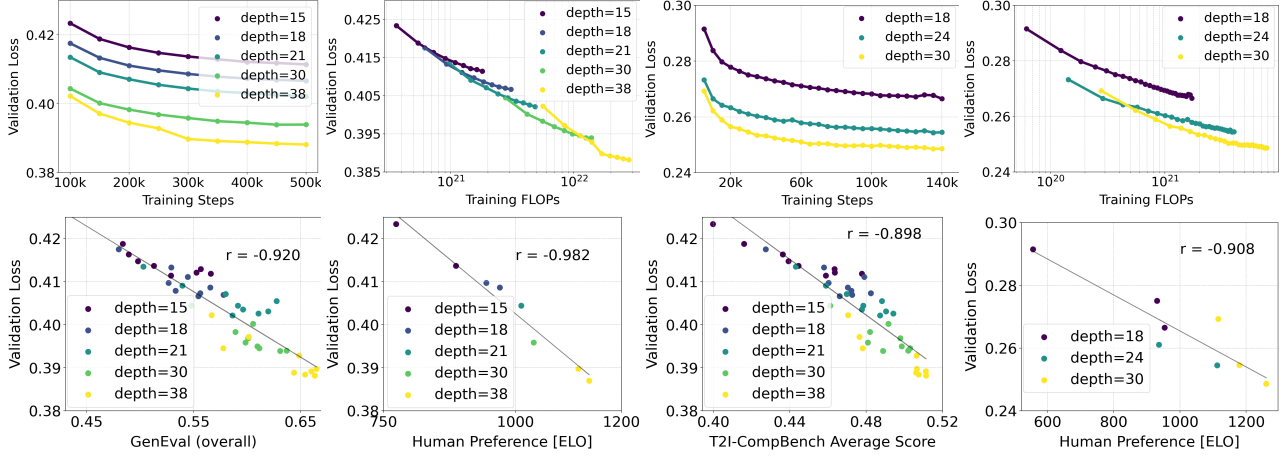


图 8. Quantitative effects of scaling. We analyze the impact of model size on performance, maintaining consistent training hyperparameters throughout. An exception is depth=38, where learning rate adjustments at 3×10^5 steps were necessary to prevent divergence. (Top) Validation loss smoothly decreases 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 等, 2023), column 1, human preference, column 2, and T2I-CompBench (Huang 等, 2023), column 3. For video models we observe a similar correlation between validation loss and human preference, column 4. . 缩放的定量效果。我们分析了模型大小对性能的影响，在整个过程中保持一致的训练超参数。深度为 38 的例外情况，在 3×10^5 步处进行了学习率调整以防止发散。（顶部）对于图像（第 1 和 2 列）和视频模型（第 3 和 4 列），验证损失随着模型大小和训练步数的增加而平稳下降。（底部）验证损失是整体模型性能的强预测器。验证损失与整体图像评估指标之间存在显著相关性，包括 GenEval (Ghosh 等, 2023)、第 1 列、人类偏好、第 2 列和 T2I-CompBench (Huang 等, 2023)、第 3 列。对于视频模型，我们观察到验证损失与人类偏好（第 4 列）之间的类似相关性。 .

2x 时间 patch。我们遵循 Blattmann 等 (2023b)，通过将时间折叠到批轴来为预训练模型输入数据。在每个注意力层中，我们重新排列视觉流中的表示，并在空间注意力操作之后、最终前馈层之前添加对所有时空 token 的完整注意力。我们的视频模型在包含 16 帧、 256^2 像素的视频上训练 140k 步，批大小为 512。我们每 5k 步在 Kinetics 数据集 (Carreira & Zisserman, 2018) 上报告验证损失。注意我们在 Figure 8 中报告的视频训练 FLOPs 仅来自视频训练，不包括图像预训练的 FLOPs。

For both the image and video domains, we observe a smooth decrease in the validation loss when increasing model size and training steps. We find the validation loss to be highly correlated to comprehensive evaluation metrics (CompBench (Huang 等, 2023), GenEval (Ghosh 等, 2023)) and to human preference. These results support the validation loss as a simple and general measure of model performance. Our results do not show saturation neither for image not for video models.

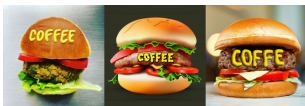
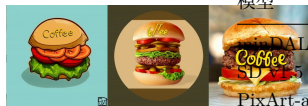
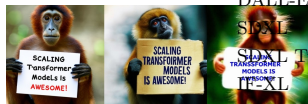
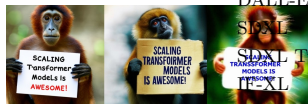
在图像和视频两个领域，我们观察到增加模型大

小和训练步数时验证损失平稳下降。我们发现验证损失与全面评价指标 (CompBench (Huang 等, 2023)、GenEval (Ghosh 等, 2023)) 和人类偏好高度相关。这些结果支持将验证损失作为模型性能的简单通用度量。我们的结果对于图像和视频模型都没有显示饱和。

Figure 12 illustrates how training a larger model for longer impacts sample quality. Tab. 5 shows the results of GenEval in full. When applying the methods presented in Section 5.3.2 and increasing training image resolution, our biggest model excels in most categories and outperforms DALL-E 3 (Betker 等, 2023), the current state of the art in prompt comprehension, in overall score.

Figure 12 说明了训练更大模型更长时间如何影响样本质量。表 5 完整展示了 GenEval 的结果。应用 Section 5.3.2 中提出的方法并增加训练图像分辨率时，我们最大的模型在大多数类别中表现卓越，在总体分数上超过 DALL-E 3 (Betker 等, 2023)，即提示理解方面的当前最先进方法。

我们的 $d = 38$ 模型在 Parti-prompts 基准 (Yu 等, 2022) 的人类偏好评估中，在视觉美学、提示遵

All text-encoders			w/o T5 (Raffel 等, 2019)			物体							
			模型			总体	单个	两个	计数	颜色	位置	颜色属性	
						DALL-E	0.23	0.73	0.11	0.12	0.37	0.02	0.01
						PixArt-alpha	0.43	0.97	0.38	0.35	0.76	0.04	0.06
						SD v2.1	0.48	0.98	0.50	0.44	0.80	0.08	0.07
“A burger patty, with the bottom bun and lettuce and tomatoes. ”COFFEE” written on it in mustard”						DALL-E 2	0.50	0.98	0.51	0.44	<u>0.85</u>	0.07	0.17
						SDXL Turbo	0.52	0.94	0.66	0.49	0.77	0.10	0.19
						SDXL	0.55	0.98	0.74	0.39	<u>0.85</u>	0.15	0.23
						DALL-E 3	0.55	1.00	0.72	0.49	0.80	0.10	0.18
							0.61	0.97	0.74	<i>0.66</i>	0.81	0.13	0.35
“A monkey holding a sign reading ”Scaling transformer models is awesome!”							0.67	0.96	<u>0.87</u>	0.47	<i>0.83</i>	0.43	<i>0.45</i>
						Ours (depth=18), 512 ²	0.58	0.97	0.72	0.52	0.78	0.16	0.34
						Ours (depth=30), 512 ²	0.64	0.96	0.80	0.65	0.73	0.33	0.37
						Ours (depth=38), 512 ²	<i>0.68</i>	0.98	0.84	<i>0.66</i>	0.74	<u>0.40</u>	0.43
						Ours (depth=38), 512 ² w/DPO	<u>0.71</u>	0.98	<u>0.89</u>	0.73	<i>0.83</i>	<i>0.34</i>	<u>0.47</u>
						Ours (depth=38), 1024 ² w/DPO	0.74	<u>0.99</u>	0.94	<u>0.72</u>	0.89	0.33	0.60

	relative CLIP score decrease [%]			路径长度
	5/50 steps	10/50 steps	20/50 steps	
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

表 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 **模型规模对采样效率的影响**. 表格展示了使用 50 采样步数和固定随机种子评估的 CLIP 分数的相对性能下降。较大的模型可以用更少的步数采样，我们将其归因于增强的鲁棒性和更好地拟合修正流模型的直路目标，这导致了更短的路径长度。路径长度通过对 50 步内的 $\|v_\theta \cdot dt\|$ 求和计算。

ating written text are more significant (38% win rate).

虽然使用多个文本编码器的主要动机是提升整体模型性能 (Balaji 等, 2022), 我们现在展示这一选择还增加了我们基于 *MM-DiT* 修正流在推理过程中的灵活性。如 Appendix B.3 所述, 我们用三个文本编码器训练我们的模型, 个别退出率为 46.3%。因此, 在推理时, 我们可以使用所有三个文本编码器的任意子集。这提供了一种平衡模型性能和改进内存效率的手段, 这对于需要大量 VRAM 的 T5-XXL (Raffel 等, 2019) 的 4.7B 参数尤其相关。有趣的是, 当仅对文本提示使用两个基于 CLIP 的文本编码器, 并用零替换 T5 嵌入时, 我们观察到性能下降有限。我们在 Figure 9 中提供定性可视化。仅对于涉及高度详细场景描述或大量书面文本的复杂提示, 我们发现使用所有三个文本编码器时的显著性能收益。这些观察也在 Figure 7 (*Ours w/o T5*) 的人类偏好评估结果中得到验证。移除 T5 对美学质量评分 (50% 胜率) 没有影响, 仅对提示遵循 (46% 胜率) 有小的影响, 而其对生成书面文本能力的贡献更显著 (38% 胜率)。

6. 结论

In this work, we presented a scaling analysis of rectified flow models for text-to-image synthesis. We proposed a novel timestep sampling for rectified flow training that improves over previous diffusion training formulations for

latent diffusion models and retains the favourable properties of rectified flows in the few-step sampling regime. We also demonstrated the advantages of our transformer-based *MM-DiT* architecture that takes the multi-modal nature of the text-to-image task into account. Finally, we performed a scaling study of this combination up to a model size of 8B parameters and 5×10^{22} training FLOPs. We showed that validation loss improvements correlate with both existing text-to-image benchmarks as well as human preference evaluations. This, in combination with our improvements in generative modeling and scalable, multimodal architectures achieves performance that is competitive with state-of-the-art proprietary models. The scaling trend shows no signs of saturation, which makes us optimistic that we can continue to improve the performance of our models in the future.

本工作分析了文本到图像合成中修正流模型的缩放特性。我们提出新的时间步采样方法用于修正流训练, 性能优于潜在扩散的先前方法, 且在少步采样中保持修正流的优良特性。我们还展示了基于 Transformer 的 *MM-DiT* 架构的优势, 该架构为多模态文本到图像任务设计。最后, 我们对该架构进行了缩放研究, 模型规模达到 8B 参数、 5×10^{22} 训练 FLOPs。验证损失的改进与现有文本到图像基准和人类偏好评估均有相关性。结合生成建模和多模态架构的创新, 我们的模型性能与最先进的专有模型相当。缩放趋势尚未饱和, 我们对继续改进模型性能充满信心。

更广泛的影响

This paper presents work whose goal is to advance the field of machine learning in general and image synthesis in particular. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here. For an extensive discussion of the general ramifications of diffusion models, we point interested readers towards (Po 等, 2023).

本文的目标是推动机器学习领域、尤其是图像合成方向的发展。我们的工作可能带来诸多社会影响, 但我们认为其中没有哪一项需要在此特别强调。关于扩散模型影响的全面讨论, 感兴趣的读者可参阅 (Po 等, 2023)。

参考文献

- Ideogram v1.0 announcement, 2024. URL <https://about.ideogram.ai/1.0>.
- Playground v2.5 announcement, 2024. URL <https://blog.playgroundai.com/playground-v2-5/>.
- Albergo, M. S. and Vanden-Eijnden, E. Building normalizing flows with stochastic interpolants, 2022.
- Atchison, J. and Shen, S. M. Logistic-normal distributions: Some properties and uses. *Biometrika*, 67(2): 261–272, 1980.
- autofaiss. autofaiss, 2023. URL <https://github.com/criteo/autofaiss>.
- Balaji, Y., Nah, S., Huang, X., Vahdat, A., Song, J., Zhang, Q., Kreis, K., Aittala, M., Aila, T., Laine, S., Catanzaro, B., Karras, T., and Liu, M.-Y. ediff-i: Text-to-image diffusion models with an ensemble of expert denoisers, 2022.
- Betker, J., Goh, G., Jing, L., Brooks, T., Wang, J., Li, L., Ouyang, L., Zhuang, J., Lee, J., Guo, Y., et al. Improving image generation with better captions. *Computer Science*. <https://cdn.openai.com/papers/dall-e-3.pdf>, 2(3), 2023.
- Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023a.
- Blattmann, A., Rombach, R., Ling, H., Dockhorn, T., Kim, S. W., Fidler, S., and Kreis, K. Align your latents: High-resolution video synthesis with latent diffusion models, 2023b.
- Brooks, T., Holynski, A., and Efros, A. A. Instruct-pix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18392–18402, 2023.
- Carlini, N., Hayes, J., Nasr, M., Jagielski, M., Sehwag, V., Tramèr, F., Balle, B., Ippolito, D., and Wallace, E. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 5253–5270, 2023.
- Carreira, J. and Zisserman, A. Quo vadis, action recognition? a new model and the kinetics dataset, 2018.
- Changpinyo, S., Sharma, P. K., Ding, N., and Soricut, R. Conceptual 12m: Pushing web-scale image-text pre-training to recognize long-tail visual concepts. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3557–3567, 2021. URL <https://api.semanticscholar.org/CorpusID:231951742>.
- Chen, D., Chou, C., Xu, Y., and Hseu, J. Bfloat16: The secret to high performance on cloud tpus, 2019. URL <https://cloud.google.com/blog/products/ai-machine-learning/bfloat16-the-secret-to-high-performance-on-cloud-tpus?hl=en>.
- Chen, J., Yu, J., Ge, C., Yao, L., Xie, E., Wu, Y., Wang, Z., Kwok, J., Luo, P., Lu, H., and Li, Z. Pixart-a: Fast training of diffusion transformer for photorealistic text-to-image synthesis, 2023.
- Chen, T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. Neural ordinary differential equations. In *Neural Information Processing Systems*, 2018. URL <https://api.semanticscholar.org/CorpusID:49310446>.
- Cherti, M., Beaumont, R., Wightman, R., Wortsman, M., Ilharco, G., Gordon, C., Schuhmann, C., Schmidt, L., and Jitsev, J. Reproducible scaling laws for contrastive language-image learning. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2023. doi: 10.1109/cvpr52729.2023.00276. URL <http://dx.doi.org/10.1109/CVPR52729.2023.00276>.

- Dai, X., Hou, J., Ma, C.-Y., Tsai, S., Wang, J., Wang, R., Zhang, P., Vandenhende, S., Wang, X., Dubey, A., Yu, M., Kadian, A., Radenovic, F., Mahajan, D., Li, K., Zhao, Y., Petrovic, V., Singh, M. K., Motwani, S., Wen, Y., Song, Y., Sumbaly, R., Ramanathan, V., He, Z., Vajda, P., and Parikh, D. Emu: Enhancing image generation models using photogenic needles in a haystack, 2023.
- Dao, Q., Phung, H., Nguyen, B., and Tran, A. Flow matching in latent space, 2023.
- Dehghani, M., Djolonga, J., Mustafa, B., Padlewski, P., Heek, J., Gilmer, J., Steiner, A., Caron, M., Geirhos, R., Alabdulmohsin, I., Jenatton, R., Beyer, L., Tschannen, M., Arnab, A., Wang, X., Riquelme, C., Minderer, M., Puigcerver, J., Evci, U., Kumar, M., van Steenkiste, S., Elsayed, G. F., Mahendran, A., Yu, F., Oliver, A., Huot, F., Bastings, J., Collier, M. P., Gritsenko, A., Birodkar, V., Vasconcelos, C., Tay, Y., Mensink, T., Kolesnikov, A., Pavetić, F., Tran, D., Kipf, T., Lučić, M., Zhai, X., Keysers, D., Harmsen, J., and Houlsby, N. Scaling vision transformers to 22 billion parameters, 2023.
- Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis, 2021.
- Dockhorn, T., Vahdat, A., and Kreis, K. Score-based generative modeling with critically-damped langevin diffusion. *arXiv preprint arXiv:2112.07068*, 2021.
- Dockhorn, T., Vahdat, A., and Kreis, K. Genie: Higher-order denoising diffusion solvers, 2022.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2020.
- Esser, P., Chiu, J., Atighehchian, P., Granskog, J., and Germanidis, A. Structure and content-guided video synthesis with diffusion models, 2023.
- Euler, L. *Institutionum calculi integralis*. Number Bd. 1 in *Institutionum calculi integralis*. imp. Acad. imp. Saent., 1768. URL <https://books.google.de/books?id=Vg80AAAAQAAJ>.
- Fischer, J. S., Gui, M., Ma, P., Stracke, N., Baumann, S. A., and Ommer, B. Boosting latent diffusion with flow matching. *arXiv preprint arXiv:2312.07360*, 2023.
- Ghosh, D., Hajishirzi, H., and Schmidt, L. Geneval: An object-focused framework for evaluating text-to-image alignment. *arXiv preprint arXiv:2310.11513*, 2023.
- Gupta, A., Yu, L., Sohn, K., Gu, X., Hahn, M., Fei-Fei, L., Essa, I., Jiang, L., and Lezama, J. Photorealistic video generation with diffusion models, 2023.
- Hessel, J., Holtzman, A., Forbes, M., Le Bras, R., and Choi, Y. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.595. URL <http://dx.doi.org/10.18653/v1/2021.emnlp-main.595>.
- Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2017.
- Ho, J. and Salimans, T. Classifier-free diffusion guidance, 2022.
- Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models, 2020.
- Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D. P., Poole, B., Norouzi, M., Fleet, D. J., and Salimans, T. Imagen video:

- High definition video generation with diffusion models, 2022.
- Hoogeboom, E., Heek, J., and Salimans, T. Simple diffusion: End-to-end diffusion for high resolution images, 2023.
- Huang, K., Sun, K., Xie, E., Li, Z., and Liu, X. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. *arXiv preprint arXiv:2307.06350*, 2023.
- Hyvärinen, A. Estimation of non-normalized statistical models by score matching. *J. Mach. Learn. Res.*, 6:695–709, 2005. URL <https://api.semanticscholar.org/CorpusID:1152227>.
- Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models, 2020.
- Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. *ArXiv*, abs/2206.00364, 2022. URL <https://api.semanticscholar.org/CorpusID:249240415>.
- Karras, T., Aittala, M., Lehtinen, J., Hellsten, J., Aila, T., and Laine, S. Analyzing and improving the training dynamics of diffusion models. *arXiv preprint arXiv:2312.02696*, 2023.
- Kingma, D. P. and Gao, R. Understanding diffusion objectives as the elbo with simple data augmentation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- Lee, K., Ippolito, D., Nystrom, A., Zhang, C., Eck, D., Callison-Burch, C., and Carlini, N. Deduplicating training data makes language models better. *arXiv preprint arXiv:2107.06499*, 2021.
- Lee, S., Kim, B., and Ye, J. C. Minimizing trajectory curvature of ode-based generative models, 2023.
- Lin, S., Liu, B., Li, J., and Yang, X. Common diffusion noise schedules and sample steps are flawed. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 5404–5411, 2024.
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. *Microsoft COCO: Common Objects in Context*, pp. 740–755. Springer International Publishing, 2014. ISBN 9783319106021. doi: 10.1007/978-3-319-10602-1_48. URL http://dx.doi.org/10.1007/978-3-319-10602-1_48.
- Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=PqvMRDCJT9t>.
- Liu, X., Gong, C., and Liu, Q. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022.
- Liu, X., Zhang, X., Ma, J., Peng, J., and Liu, Q. InstafLOW: One step is enough for high-quality diffusion-based text-to-image generation, 2023.
- Loshchilov, I. and Hutter, F. Fixing weight decay regularization in adam. *ArXiv*, abs/1711.05101, 2017. URL <https://api.semanticscholar.org/CorpusID:3312944>.
- Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models, 2023.
- Ma, N., Goldstein, M., Albergo, M. S., Boffi, N. M., Vanden-Eijnden, E., and Xie, S. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers, 2024.
- Nichol, A. Dall-e 2 pre-training mitigations. <https://openai.com/research/dall-e-2-pre-training-mitigations>, 2022.

- Nichol, A. and Dhariwal, P. Improved denoising diffusion probabilistic models, 2021.
- NovelAI. Novelai improvements on stable diffusion, 2022. URL <https://blog.novelai.net/novelai-improvements-on-stable-diffusion-e10d38db82ac>.
- Peebles, W. and Xie, S. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2023. doi: 10.1109/iccv51070.2023.00387. URL <http://dx.doi.org/10.1109/ICCV51070.2023.00387>.
- Pernias, P., Rampas, D., Richter, M. L., Pal, C. J., and Aubreville, M. Wuerstchen: An efficient architecture for large-scale text-to-image diffusion models, 2023.
- Pizzi, E., Roy, S. D., Ravindra, S. N., Goyal, P., and Douze, M. A self-supervised descriptor for image copy detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14532–14542, 2022.
- Po, R., Yifan, W., Golyanik, V., Aberman, K., Barron, J. T., Bermano, A. H., Chan, E. R., Dekel, T., Holynski, A., Kanazawa, A., et al. State of the art on diffusion models for visual computing. *arXiv preprint arXiv:2310.07204*, 2023.
- Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., and Rombach, R. Sdxl: Improving latent diffusion models for high-resolution image synthesis, 2023.
- Pooladian, A.-A., Ben-Hamu, H., Domingo-Enrich, C., Amos, B., Lipman, Y., and Chen, R. T. Q. Multisample flow matching: Straightening flows with minibatch couplings, 2023.
- Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision, 2021.
- Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. *arXiv:2305.18290*, 2023.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer, 2019.
- Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents, 2022.
- Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022. doi: 10.1109/cvpr52688.2022.01042. URL <http://dx.doi.org/10.1109/CVPR52688.2022.01042>.
- Ronneberger, O., Fischer, P., and Brox, T. *U-Net: Convolutional Networks for Biomedical Image Segmentation*, pp. 234–241. Springer International Publishing, 2015. ISBN 9783319245744. doi: 10.1007/978-3-319-24574-4_28. URL http://dx.doi.org/10.1007/978-3-319-24574-4_28.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. S., Berg, A. C., and Fei-Fei, L. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211 – 252, 2014. URL <https://api.semanticscholar.org/CorpusID:2930547>.
- Saharia, C., Chan, W., Chang, H., Lee, C., Ho, J., Salimans, T., Fleet, D., and Norouzi, M. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings*, pp. 1–10, 2022a.

- Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S. K. S., Ayan, B. K., Mahdavi, S. S., Lopes, R. G., Salimans, T., Ho, J., Fleet, D. J., and Norouzi, M. Photorealistic text-to-image diffusion models with deep language understanding, 2022b.
- Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D. J., and Norouzi, M. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4713–4726, 2022c.
- Sauer, A., Chitta, K., Müller, J., and Geiger, A. Projected gans converge faster. *Advances in Neural Information Processing Systems*, 2021.
- Sauer, A., Lorenz, D., Blattmann, A., and Rombach, R. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042*, 2023.
- Sheynin, S., Polyak, A., Singer, U., Kirstain, Y., Zohar, A., Ashual, O., Parikh, D., and Taigman, Y. Emu edit: Precise image editing via recognition and generation tasks. *arXiv preprint arXiv:2311.10089*, 2023.
- Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., Zhang, S., Hu, Q., Yang, H., Ashual, O., Gafni, O., Parikh, D., Gupta, S., and Taigman, Y. Make-a-video: Text-to-video generation without text-video data, 2022.
- Sohl-Dickstein, J. N., Weiss, E. A., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. *ArXiv*, abs/1503.03585, 2015. URL <https://api.semanticscholar.org/CorpusID:14888175>.
- Somepalli, G., Singla, V., Goldblum, M., Geiping, J., and Goldstein, T. Diffusion art or digital forgery? investigating data replication in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6048–6058, 2023a.
- Somepalli, G., Singla, V., Goldblum, M., Geiping, J., and Goldstein, T. Understanding and mitigating copying in diffusion models. *arXiv preprint arXiv:2305.20086*, 2023b.
- Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models, 2022.
- Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution, 2020.
- Song, Y., Sohl-Dickstein, J. N., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *ArXiv*, abs/2011.13456, 2020. URL <https://api.semanticscholar.org/CorpusID:227209335>.
- Tong, A., Malkin, N., Huguet, G., Zhang, Y., Rector-Brooks, J., Fatras, K., Wolf, G., and Bengio, Y. Improving and generalizing flow-based generative models with minibatch optimal transport, 2023.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need, 2017.
- Villani, C. Optimal transport: Old and new. 2008. URL <https://api.semanticscholar.org/CorpusID:118347220>.
- Vincent, P. A connection between score matching and denoising autoencoders. *Neural Computation*, 23: 1661–1674, 2011. URL <https://api.semanticscholar.org/CorpusID:5560643>.
- Wallace, B., Dang, M., Rafailov, R., Zhou, L., Lou, A., Purushwalkam, S., Ermon, S., Xiong, C., Joty, S., and Naik, N. Diffusion Model Alignment Using Direct Preference Optimization. *arXiv:2311.12908*, 2023.
- Wang, W., Lv, Q., Yu, W., Hong, W., Qi, J., Wang, Y., Ji, J., Yang, Z., Zhao, L., Song, X., et al. Cogvlm:

Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*, 2023.

Wortsman, M., Liu, P. J., Xiao, L., Everett, K., Alemi, A., Adlam, B., Co-Reyes, J. D., Gur, I., Kumar, A., Novak, R., Pennington, J., Sohl-dickstein, J., Xu, K., Lee, J., Gilmer, J., and Kornblith, S. Small-scale proxies for large-scale transformer training instabilities, 2023.

Yu, J., Xu, Y., Koh, J. Y., Luong, T., Baid, G., Wang, Z., Vasudevan, V., Ku, A., Yang, Y., Ayan, B. K., et al. Scaling Autoregressive Models for Content-Rich Text-to-Image Generation. *arXiv:2206.10789*, 2022.

Zhai, X., Kolesnikov, A., Houlsby, N., and Beyer, L. Scaling vision transformers. In *CVPR*, pp. 12104–12113, 2022.

Zhang, B. and Sennrich, R. Root mean square layer normalization, 2019.

Supplementary

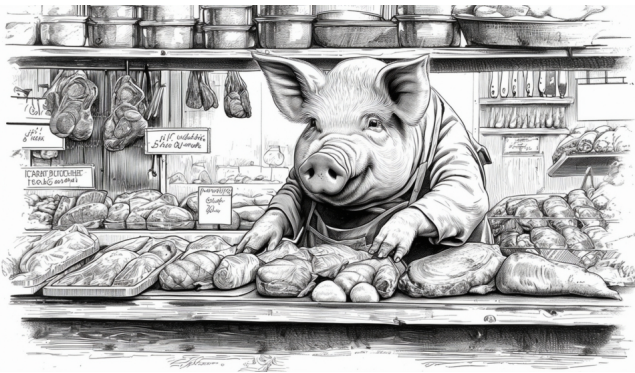
A. 背景

Diffusion Models (Sohl-Dickstein 等, 2015; Song 等, 2020; Ho 等, 2020) generate data by approximating the reverse ODE to a stochastic forward process which transforms data to noise. They have become the standard approach for generative modeling of images (Dhariwal & Nichol, 2021; Ramesh 等, 2022; Saharia 等, 2022b; Rombach 等, 2022; Balaji 等, 2022) and videos (Singer 等, 2022; Ho 等, 2022; Esser 等, 2023; Blattmann 等, 2023b; Gupta 等, 2023). Since these models can be derived both via a variational lower bound on the negative likelihood (Sohl-Dickstein 等, 2015) and score matching (Hyvärinen, 2005; Vincent, 2011; Song & Ermon, 2020),

various formulations of forward- and reverse processes (Song 等, 2020; Dockhorn 等, 2021), model parameterizations (Ho 等, 2020; Ho & Salimans, 2022; Karras 等, 2022), loss weightings (Ho 等, 2020; Karras 等, 2022) and ODE solvers (Song 等, 2022; Lu 等, 2023; Dockhorn 等, 2022) have led to a large number of different training objectives and sampling procedures. More recently, the seminal works of Kingma & Gao (2023) and Karras 等 (2022) have proposed unified formulations and introduced new theoretical and practical insights for training (Karras 等, 2022; Kingma & Gao, 2023) and inference (Karras 等, 2022). However, despite these improvements, the trajectories of common ODEs involve partly significant amounts of curvature (Karras 等, 2022; Liu 等, 2022), which requires increased amounts of solver steps and, thus, renders fast inference difficult. To overcome this, we adopt rectified flow models whose formulation allows for learning straight ODE trajectories.

扩散模型 (Sohl-Dickstein 等, 2015; Song 等, 2020; Ho 等, 2020) 通过近似随机前向过程的反向常微分方程来生成数据, 该过程将数据转换为噪声。这些模型已成为图像 (Dhariwal & Nichol, 2021; Ramesh 等, 2022; Saharia 等, 2022b; Rombach 等, 2022; Balaji 等, 2022) 和视频 (Singer 等, 2022; Ho 等, 2022; Esser 等, 2023; Blattmann 等, 2023b; Gupta 等, 2023) 生成建模的标准方法。这些模型可以通过负对数似然的变分下界 (Sohl-Dickstein 等, 2015) 和分数匹配 (Hyvärinen, 2005; Vincent, 2011; Song & Ermon, 2020) 两种方式推导。前向和反向过程 (Song 等, 2020; Dockhorn 等, 2021)、模型参数化 (Ho 等, 2020; Ho & Salimans, 2022; Karras 等, 2022)、损失加权 (Ho 等, 2020; Karras 等, 2022) 和常微分方程求解器 (Song 等, 2022; Lu 等, 2023; Dockhorn 等, 2022) 的多种公式导致了大量不同的训练目标和采样过程。最近, Kingma & Gao (2023) 和 Karras 等 (2022) 的开创性工作提出了统一公式, 为训练 (Karras 等, 2022; Kingma & Gao, 2023) 和推理 (Karras 等, 2022) 引入了新的理论和实践见解。然而, 尽管有这些改进, 常见常微分方程的轨迹涉及相当的曲率 (Karras 等, 2022; Liu 等, 2022), 这需要增加求解器步数, 因此使快速推理变得困难。为了克服这一点, 我们采用修正流模型, 其公式允许学习直线常微分方程轨迹。

Rectified Flow Models (Liu 等, 2022; Albergo & Vanden-Eijnden, 2022; Lipman 等, 2023) approach generative modeling by constructing a transport map between two distributions through an ordinary differential equation (ODE). This approach has close connections to continuous normalizing flows (CNF) (Chen 等, 2018) as well as diffusion models. Compared to CNFs, Rectified Flows and Stochastic Interpolants have the advantage that they do not require simulation of the ODE during training. Compared to diffusion models, they can result in ODEs that are faster to simulate than the probability flow ODE (Song 等, 2020) associated with diffusion models. Nevertheless, they do not result in optimal transport solutions, and multiple works aim to minimize the trajectory curvature further (Lee 等, 2023; Tong 等, 2023; Pooladian 等, 2023). (Dao 等, 2023; Ma 等, 2024) demonstrate the feasibility of rectified flow formulations for class-conditional image synthesis, (Fischer 等, 2023) for latent-space upsampling, and (Liu 等, 2023) apply the reflow procedure of (Liu 等, 2022) to distill a pretrained text-to-image model (Rombach 等, 2022). Here, we are interested in rectified flows as the foundation for text-to-image synthesis with fewer sampling steps. We perform an extensive comparison between different formulations and loss weightings and propose a new timestep schedule for training of rectified flows with improved performance.



Detailed pen and ink drawing of a happy pig butcher selling meat in its shop.



a massive alien space ship that is shaped like a pretzel.



A kangaroo holding a beer, wearing ski goggles and passionately singing silly songs.



An entire universe inside a bottle sitting on the shelf at walmart on sale.



A cheesburger surfing the vibe wave at night



A swamp ogre with a pearl earring by Johannes Vermeer



A car made out of vegetables.



heat death of the universe, line art



A crab made of cheese on a plate



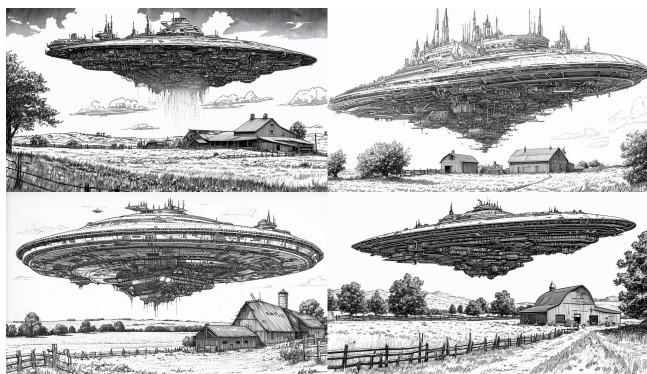
Dystopia of thousand of workers picking cherries and feeding them into a machine that runs on steam and is as large as a skyscraper. Written on the side of the machine: "SD3 Paper"



28

translucent pig, inside is a smaller pig.





detailed pen and ink drawing of a massive complex alien space ship above a farm in the middle of nowhere.



photo of a bear wearing a suit and top hat in a river in the middle of a forest holding a sign that says "I can't bear it".



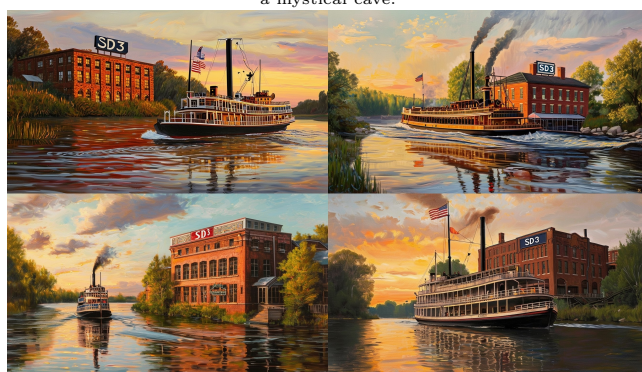
tilt shift aerial photo of a cute city made of sushi on a wooden table in the evening.



dark high contrast render of a psychedelic tree of life illuminating dust in a mystical cave.



an anthropomorphic fractal person behind the counter at a fractal themed restaurant.



beautiful oil painting of a steamboat in a river in the afternoon. On the side of the river is a large brick building with a sign on top that says SD3.



an anthropomorphic pink donut with a mustache and cowboy hat standing in a log cabin in a forest with an old 1970s orange truck in the driveway



fox sitting in front of a computer in a messy room at night. On the screen is a 3d modeling program with a line render of a zebra.

修正流模型 (Liu 等, 2022; Albergo & Vanden-Eijnden, 2022; Lipman 等, 2023) 通过构造两个分布之间的常微分方程传输映射来进行生成建模。这种方法与连续归一化流 (Chen 等, 2018) 和扩散模型密切相关。与连续归一化流相比, 修正流和随机插值的优点是在训练过程中无需模拟常微分方程。与扩散模型相比, 它们可以生成常微分方程, 其模拟速度比扩散模型相关的概率流常微分方程 (Song 等, 2020) 更快。但它们不产生最优传输解。多项工作着眼于进一步降低轨迹曲率 (Lee 等, 2023; Tong 等, 2023; Pooladian 等, 2023)。 (Dao 等, 2023; Ma 等, 2024) 证明了修正流公式在类条件图像合成中的可行性, (Fischer 等, 2023) 在潜在空间上采样中的可行性, (Liu 等, 2023) 应用 (Liu 等, 2022) 的重流过程蒸馏预训练的文本到图像模型 (Rombach 等, 2022)。我们关注修正流作为少步采样文本到图像合成的基础。我们对不同公式和损失加权进行了广泛对比, 提出了改进修正流训练性能的新时间步调度。

Scaling Diffusion Models The transformer architecture (Vaswani 等, 2017) is well known for its scaling properties in NLP (Kaplan 等, 2020) and computer vision tasks (Dosovitskiy 等, 2020; Zhai 等, 2022). For diffusion models, U-Net architectures (Ronneberger 等, 2015) have been the dominant choice (Ho 等, 2020; Rombach 等, 2022; Balaji 等, 2022). While some recent works explore diffusion transformer backbones (Peebles & Xie, 2023; Chen 等, 2023; Ma 等, 2024), scaling laws for text-to-image diffusion models remain unexplored.

缩放扩散模型 Transformer 架构 (Vaswani 等, 2017) 因其在自然语言处理 (Kaplan 等, 2020) 和计算机视觉任务 (Dosovitskiy 等, 2020; Zhai 等, 2022) 中的缩放特性而众所周知。对于扩散模型, U-Net 架构 (Ronneberger 等, 2015) 一直是主流选择 (Ho 等, 2020; Rombach 等, 2022; Balaji 等, 2022)。虽然最近的一些工作探索扩散 Transformer 骨干网络 (Peebles & Xie, 2023; Chen 等, 2023; Ma 等, 2024), 但文本到图像扩散模型的缩放定律仍未被探索。

B. 关于流匹配

B.1. 免仿真流训练的细节

Following (Lipman 等, 2023), to see that $u_t(z)$ generates p_t , we note that the continuity equation provides a necessary and sufficient condition (Villani, 2008):

遵循 (Lipman 等, 2023), 为说明 $u_t(z)$ 生成 p_t , 我们注意到连续性方程给出了一个充要条件 (Villani, 2008):

$$\frac{d}{dt}p_t(x) + \nabla \cdot [p_t(x)v_t(x)] = 0 \leftrightarrow v_t \text{ 生成概率密度路径 } p_t. \quad (25)$$

Therefore it suffices to show that

因此只需证明:

$$-\nabla \cdot [u_t(z)p_t(z)] = -\nabla \cdot [\mathbb{E}_{\epsilon \sim \mathcal{N}(0, I)} u_t(z|\epsilon) \frac{p_t(z|\epsilon)}{p_t(z)} p_t(z)] \quad (26)$$

$$= \mathbb{E}_{\epsilon \sim \mathcal{N}(0, I)} -\nabla \cdot [u_t(z|\epsilon)p_t(z|\epsilon)] \quad (27)$$

$$= \mathbb{E}_{\epsilon \sim \mathcal{N}(0, I)} \frac{d}{dt}p_t(z|\epsilon) = \frac{d}{dt}p_t(z), \quad (28)$$

where we used the continuity equation Equation (25) for $u_t(z|\epsilon)$ in line Equation (27) to Equation (28) since $u_t(z|\epsilon)$ generates $p_t(z|\epsilon)$ and the definition of Equation (6) in line Equation (26)

其中, 在 Equation (27) 到 Equation (28) 一行, 我们对 $u_t(z|\epsilon)$ 使用了连续性方程 Equation (25) (因为 $u_t(z|\epsilon)$ 生成 $p_t(z|\epsilon)$); 在 Equation (26) 一行使用了 Equation (6) 的定义。

The equivalence of objectives $\mathcal{L}_{FM} \Leftrightarrow \mathcal{L}_{CFM}$ (Lipman 等, 2023) follows from

目标 $\mathcal{L}_{FM} \Leftrightarrow \mathcal{L}_{CFM}$ 的等价性 (Lipman 等, 2023) 可由下式推出:

$$\mathcal{L}_{FM}(\Theta) = \mathbb{E}_{t, p_t(z)} \|v_\Theta(z, t) - u_t(z)\|_2^2 \quad (29)$$

$$= \mathbb{E}_{t, p_t(z)} \|v_\Theta(z, t)\|_2^2 - 2\mathbb{E}_{t, p_t(z)} \langle v_\Theta(z, t) | u_t(z) \rangle + c \quad (30)$$

$$= \mathbb{E}_{t, p_t(z)} \|v_\Theta(z, t)\|_2^2 - 2\mathbb{E}_{t, p_t(z|\epsilon), p(\epsilon)} \langle v_\Theta(z, t) | u_t(z|\epsilon) \rangle + c \quad (31)$$

$$= \mathbb{E}_{t, p_t(z|\epsilon), p(\epsilon)} \|v_\Theta(z, t) - u_t(z|\epsilon)\|_2^2 + c' = \mathcal{L}_{CFM}(\Theta) + c' \quad (32)$$

where c, c' do not depend on Θ and line Equation (30) to line Equation (31) follows from:

其中 c, c' 不依赖于 Θ , 而从 Equation (30) 到 Equation (31) 一行由下式得到:

$$\mathbb{E}_{p_t(z|\epsilon), p(\epsilon)} \langle v_\Theta(z, t) | u_t(z|\epsilon) \rangle = \int dz \int d\epsilon p_t(z|\epsilon) p(\epsilon) \langle v_\Theta(z, t) | u_t(z|\epsilon) \rangle \quad (33)$$

$$= \int dz p_t(z) \langle v_\Theta(z, t) | \int d\epsilon \frac{p_t(z|\epsilon)}{p_t(z)} p(\epsilon) u_t(z|\epsilon) \rangle \quad (34)$$

$$= \int dz p_t(z) \langle v_\Theta(z, t) | u_t(z) \rangle = \mathbb{E}_{p_t(z)} \langle v_\Theta(z, t) | u_t(z) \rangle \quad (35)$$

where we extended with $\frac{p_t(z)}{p_t(z)}$ in line Equation (34) and used the definition of Equation (6) in line Equation (34) to Equation (35).

其中, 在 Equation (34) 一行我们乘以 $\frac{p_t(z)}{p_t(z)}$ 进行扩展, 并在 Equation (34) 到 Equation (35) 一行使用了 Equation (6) 的定义。

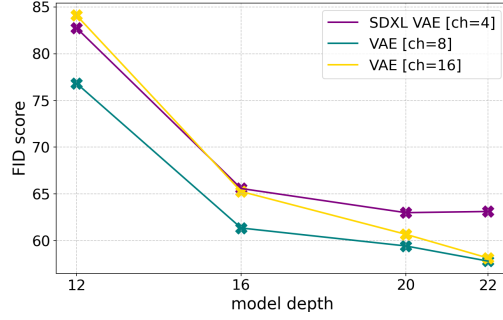


图 10. FID scores after training flow models with different sizes (parameterized via their depth) on the latent space of different autoencoders (4 latent channels, 8 channels and 16 channels) as discussed in Section 5.2.1. As expected, the flow model trained on the 16-channel autoencoder space needs more 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.

在不同大小的流模型（通过其深度参数化）在不同自编码器的潜在空间（4 个潜在通道、8 个通道和 16 个通道）上训练后的 FID 分数，如 Section 5.2.1 中所讨论的。如预期，在 16 通道自编码器空间上训练的流模型需要更大的模型容量才能达到类似的性能。在深度 $d = 22$ 时，8 通道和 16 通道之间的差距可以忽略不计。我们选择 16 通道模型，因为我们的最终目标是缩放到更大的模型大小。

B.2. 图像与文本表示的细节

Latent Image Representation We follow LDM (Rombach 等, 2022) and use a pretrained autoencoder to represent RGB images $X \in \mathbb{R}^{H \times W \times 3}$ in a smaller latent space $x = E(X) \in \mathbb{R}^{h \times w \times d}$. We use a spatial downsampling factor of 8, such that $h = \frac{H}{8}$ and $w = \frac{W}{8}$, and experiment with different values for d in Section 5.2.1. We always apply the forward process from Equation 2 in the latent space, and when sampling a representation x via Equation 1, we decode it back into pixel space $X = D(x)$ via the decoder D . We follow Rombach 等 (2022) and normalize the latents by their mean and standard deviation, which are globally computed over a subset of the training data.

潜在图像表示我们遵循 LDM (Rombach 等, 2022)，使用预训练自编码器在潜在空间中表示 RGB 图像。空间下采样因子为 8，因此 $h = \frac{H}{8}$ 和 $w = \frac{W}{8}$ 。我们在 Section 5.2.1 中对不同的 d 值进行了实验。始终在潜在空间中应用前向过程 Equation 2；采样表示 x 后通过解码器 D 解码回像素空间 $X = D(x)$ 。我们按 Rombach 等 (2022) 的方法，用训练数据集全局计算的均值和标准差归一化潜在。Figure 10 shows how generative model training for different d evolves as a function of model capacity, as discussed in Section 5.2.1.

Text Representation Similar to the encoding of images to latent representations, we also follow previous approaches (Saharia 等, 2022b; Balaji 等, 2022) and encode the text conditioning c using pretrained, frozen text models. In particular, for all experiments, we use a combination of CLIP (Radford 等, 2021) models and an encoder-decoder text model. Specifically, we encode c with the text encoders of both a CLIP L/14 model of Radford 等 (2021) as well as an OpenCLIP bigG/14 model of Cherti 等 (2023). We concatenate the pooled outputs, of sizes 768 and 1280 respectively, to obtain a vector conditioning $c_{\text{vec}} \in \mathbb{R}^{2048}$. We also concatenate the penultimate hidden representations channel-wise to a CLIP context conditioning $c_{\text{ctxt}}^{\text{CLIP}} \in \mathbb{R}^{77 \times 2048}$. Next, we encode c also to the final hidden representation, $c_{\text{ctxt}}^{\text{T5}} \in \mathbb{R}^{77 \times 4096}$, of the encoder of a T5-v1.1-XXL model (Raffel 等, 2019). Finally, we zero-pad $c_{\text{ctxt}}^{\text{CLIP}}$ along the channel axis to 4096 dimensions to match the T5 representation and concatenate it along the sequence axis with $c_{\text{ctxt}}^{\text{T5}}$ to obtain the final context representation $c_{\text{ctxt}} \in \mathbb{R}^{154 \times 4096}$. These two caption representations, c_{vec} and c_{ctxt} , are used in two different ways as described in Section 4.

文本表示类似于图像编码到潜在表示，我们采用之前的方法 (Saharia 等, 2022b; Balaji 等, 2022)，用预训练的冻结文本模型编码文本条件 c 。在所有实验中，我们结合使用 CLIP (Radford 等, 2021) 模型和编码器-解码器

器文本模型。具体地，我们用 CLIP L/14 模型 Radford 等 (2021) 和 OpenCLIP bigG/14 模型 Cherti 等 (2023) 的文本编码器对 c 编码。连接两者的池化输出（大小分别为 768 和 1280），得到向量条件 $c_{\text{vec}} \in \mathbb{R}^{2048}$ 。同时按通道方向连接倒数第二层隐藏表示得到 CLIP 上下文条件 $c_{\text{ctxt}}^{\text{CLIP}} \in \mathbb{R}^{77 \times 2048}$ 。然后将 c 编码为 T5-v1.1-XXL 模型 (Raffel 等, 2019) 编码器的最终隐藏表示 $c_{\text{ctxt}}^{\text{T5}} \in \mathbb{R}^{77 \times 4096}$ 。最后，沿通道轴将 $c_{\text{ctxt}}^{\text{CLIP}}$ 零填充至 4096 维以匹配 T5 表示，再沿序列轴与 $c_{\text{ctxt}}^{\text{T5}}$ 连接，得到最终上下文表示 $c_{\text{ctxt}} \in \mathbb{R}^{154 \times 4096}$ 。两个描述文本表示 c_{vec} 和 c_{ctxt} 按 Section 4 所述采用两种不同方式。

B.3. 对 Section 5.1 中实验的预备知识

Datasets We use two datasets to account for the missing of a standard text-to-image benchmark. As a widely used dataset, we convert the ImageNet dataset (Russakovsky 等, 2014) into a dataset suitable for text-to-image models by adding captions of the form “a photo of a `<class name>`” to images, where `<class name>` is randomly chosen from one of the provided names for the image’s class label.

As a more realistic text-to-image dataset, we use the CC12M dataset (Changpinyo 等, 2021) for training.

数据集 由于缺少标准的文本到图像基准，我们使用两个数据集。首先，我们将 ImageNet 数据集 (Russakovsky 等, 2014) 改造为文本到图像数据集。为图像添加“a photo of a `<class name>`”格式的描述文本，其中 `<class name>` 从该图像类标签的名称中随机选择。此外，我们还使用 CC12M 数据集 (Changpinyo 等, 2021) 作为更逼真的文本到图像数据集进行训练。

Optimization In this experiment, we train all models using a global batch size of 1024 using the AdamW optimizer (Loshchilov & Hutter, 2017) with a learning rate of 10^{-4} and 1000 linear warmup steps. We use mixed-precision training and keep a copy of the model weights which gets updated every 100 training batches with an exponential moving average (EMA) using a decay factor of 0.99. For unconditional diffusion guidance (Ho & Salimans, 2022), we set the outputs of each of the three text encoders independently to zero with a probability of 46.4%, such that we roughly train an unconditional model in 10% of all steps.

优化 在该实验中，我们使用 AdamW 优化器 (Loshchilov & Hutter, 2017) 训练所有模型，全局批大小为 1024，学习率为 10^{-4} ，线性预热步数为 1000。我们采用混合精度训练，并保持一份模型权重副本，该副本每 100 个训练批次用指数移动平均（EMA）更新一次，衰减因子为 0.99。对于无分类器引导 (Ho & Salimans, 2022)，我们独立地将三个文本编码器的输出归零，概率为 46.4

Evaluation

As described in Section 5.1, we use CLIP scores, FID and validation losses to evaluate our models regularly during training on the COCO-2014 validation split (Lin 等, 2014).

评估 如 Section 5.1 所述，我们在 COCO-2014 验证集上 (Lin 等, 2014) 定期使用 CLIP 分数、FID 和验证损失来评估模型。

As the loss values differ widely in magnitude and variance for different timesteps, we evaluate them in a stratified way on eight equally spaced values in the time interval $[0, 1]$.

由于不同时间步的损失值在大小和方差上差异很大，我们在时间间隔 $[0, 1]$ 内八个均匀分布的值上以分层方式评估损失。

To analyze how different approaches behave under different sampler settings, we produce 1000 samples for each of the samplers which differ in guidance scales as well as number of sampling steps. We evaluate these samples with CLIP scores using CLIP L/14 (Radford 等, 2021) and also compute FID between CLIP L/14 image features of these samples and the images of the validation set. For sampling, we always use a Euler discretization (Euler, 1768) of Equation 1 and six

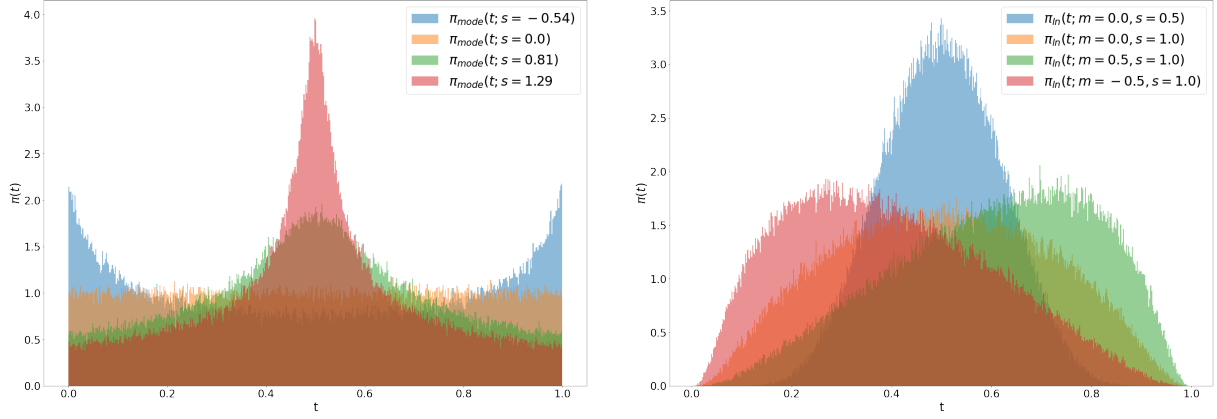


图 11. The mode (left) and logit-normal (right) distributions that we explore for biasing the sampling of training timesteps. 我们用来偏置训练时间步采样的众数（左）和 logit 正态（右）分布。



图 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.

缩放的定性效果。展示的示例展示了缩放训练步数（从左到右：50k、200k、350k、500k）和模型大小（从上到下：depth=15、30、38）对 PartiPrompts 的影响，强调了训练持续时间和模型复杂性的影响。

different settings: 50 steps with classifier-free-guidance scales 1.0, 2.5, 5.0, and 5, 10, 25 steps with classifier-free-guidance scale 5.0.

为了分析不同方法在不同采样器设置下的表现，我们为每个采样器生成 1000 个样本，这些采样器在引导强度和采样步数上有所不同。我们使用 CLIP L/14 (Radford 等, 2021) 用 CLIP 分数评估这些样本，并计算这些样本的 CLIP L/14 图像特征与验证集中图像之间的 FID。采样时，我们始终使用 Equation 1 的 Euler 离散化 (Euler, 1768) 和六种不同的设置：50 步且无分类器引导强度为 1.0、2.5、5.0 的设置，以及 5、10、25 步且无分类器引导强度为 5.0 的设置。

B.4. 改进修正流模型的信噪比采样器

如 Section 2 所述，我们为用于训练修正流模型的时间步引入了新的密度函数 $\pi(t)$ 。Figure 11 可视化了 Section 3.1 中介绍的 logit 正态采样器和众数采样器的分布。值得注意的是，如我们在 Section 5.1 中所示，logit 正态采样器相比经典的均匀修正流公式 (Liu 等, 2022) 以及已有的扩散基线（如 EDM (Karras 等, 2022) 和 LDM-Linear (Rombach 等, 2022)）的表现更优。

C. 直接偏好优化



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

基础模型和 DPO 微调模型之间的比较。DPO 微调通常产生美观程度更高且拼写更正确的样本。

Direct Preference Optimization (DPO) (Rafailov 等, 2023) is a technique to finetune LLMs with preference data. Recently, this method has been adapted to preference finetuning of text-to-image diffusion models (Wallace 等, 2023). In this section, we verify that our model is also amenable to preference optimization. In particular, we apply the method introduced in Wallace 等 (2023) to our 2B and 8B parameter base model. Rather than finetuning the entire model, we introduce learnable Low-Rank Adaptation (LoRA) matrices (of rank 128) for all linear layers as is common practice. We finetune these new parameters for 4k and 2k iteration for the 2B and 8B base model, respectively. We then evaluate the resulting model in a human preference study using a subset of 128 captions from the Partiprompts set (Yu 等, 2022) (roughly three voter per prompt and comparison). Figure 14 shows that our base models can be effectively tuned for human preference. Figure 13 shows samples of the respective base models and DPO-finetuned models.

直接偏好优化 (DPO) (Rafailov 等, 2023) 是一种使用偏好数据微调大语言模型的技术。该方法最近已被推广到文本到图像扩散模型的偏好微调 (Wallace 等, 2023)。在本节中，我们验证了我们的模型也适用于偏好优化。具体地，我们将 Wallace 等 (2023) 中介绍的方法应用于我们的 2B 和 8B 参数基模型。我们没有微调整个模型，而是为所有线性层引入了可学习的低秩适应 (LoRA) 矩阵（秩为 128），这是常见的做法。我们分别对 2B 和 8B 基模型的这些新参数进行了 4k 和 2k 次迭代的微调。随后，我们使用 Partiprompts 集 (Yu 等, 2022) 中的 128 个描述文本的子集（每个描述文本大约有三个评估者）在人类偏好研究中评估了生成的模型。Figure 14

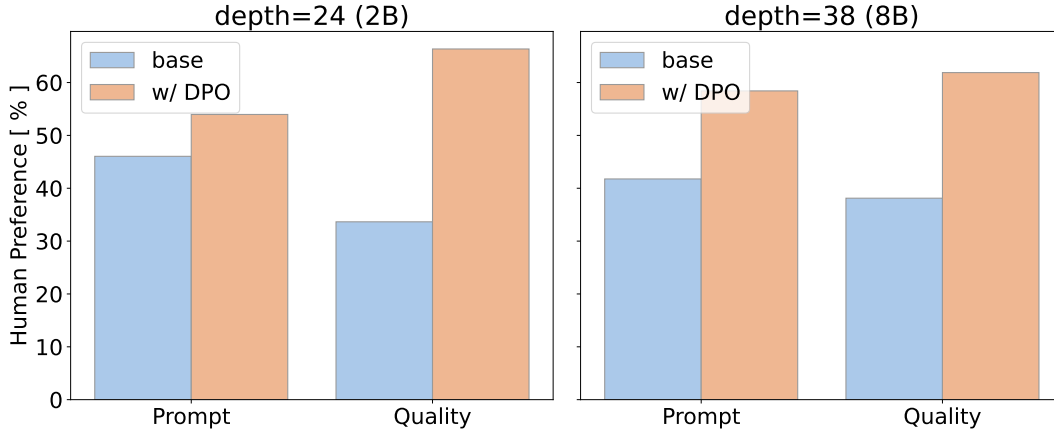


图 14. Human preference evaluation between base models and DPO-finetuned models. Human evaluators prefer DPO-finetuned models for both prompt following and general quality.

基础模型和 DPO 微调模型之间的人类偏好评估。人类评估者在提示遵循和整体质量两个方面都偏好 DPO 微调模型。

展示了我们的基模型可以有效地针对人类偏好进行调优。Figure 13 展示了各个基模型和 DPO 微调模型的样本。

D. 基于指令的图像编辑的微调

A common approach for training instruction based image editing and general image-to-image diffusion models is to concatenate the latents of the input image to the noised latents of the diffusion target along the channel dimension before feeding the input into a U-Net (Brooks 等, 2023; Sheynin 等, 2023; Saharia 等, 2022a,c). We follow the same approach, concatenating input and target along the channels before patching, and demonstrate that the same method is applicable to our proposed architecture. We finetune the 2B parameter base model on a dataset consisting of image-to-image editing tasks similar to the distribution of the InstructPix2Pix dataset (Brooks 等, 2023) as well as inpainting, segmentation, colorization, deblurring and controlnet tasks similar to Emu Edit and Palette (Sheynin 等, 2023; Saharia 等, 2022a). As shown in Fig 15 we observe that the resulting 2B Edit model has the capability to manipulate text in a given image, even though no text manipulation tasks were included in the training data. We were not able to reproduce similar results when training a SDXL-based (Podell 等, 2023) editing model on the same data.

训练基于指令的图像编辑和通用图像到图像扩散模型的常见方法是在将输入送入 U-Net 前，沿通道维度将输入图像的潜在与扩散目标的加噪潜在连接起来 (Brooks 等, 2023; Sheynin 等, 2023; Saharia 等, 2022a,c)。我们采用相同的方法，在切分前沿通道连接输入和目标，并演示了相同的方法同样适用于我们提议的架构。我们在包含图像到图像编辑任务的数据集上微调 2B 参数基模型，这些编辑任务的分布类似于 InstructPix2Pix 数据集 (Brooks 等, 2023)，以及修复、分割、着色、去模糊和 ControlNet 任务，类似于 Emu Edit 和 Palette (Sheynin 等, 2023; Saharia 等, 2022a)。如图15所示，我们观察到生成的 2B 编辑模型具有在给定的图像中处理文本的能力，尽管训练数据中没有包含文本处理任务。在使用相同数据训练基于 SDXL (Podell 等, 2023) 的编辑模型时，我们无法复现类似的结果。

E. 大规模文本到图像训练的数据预处理

E.1. 预计算图像和文本嵌入

Our model uses the output of multiple pretrained, frozen networks as inputs (autoencoder latents and text encoder representations). Since these outputs are constant during training, we precompute them once for the entire dataset.

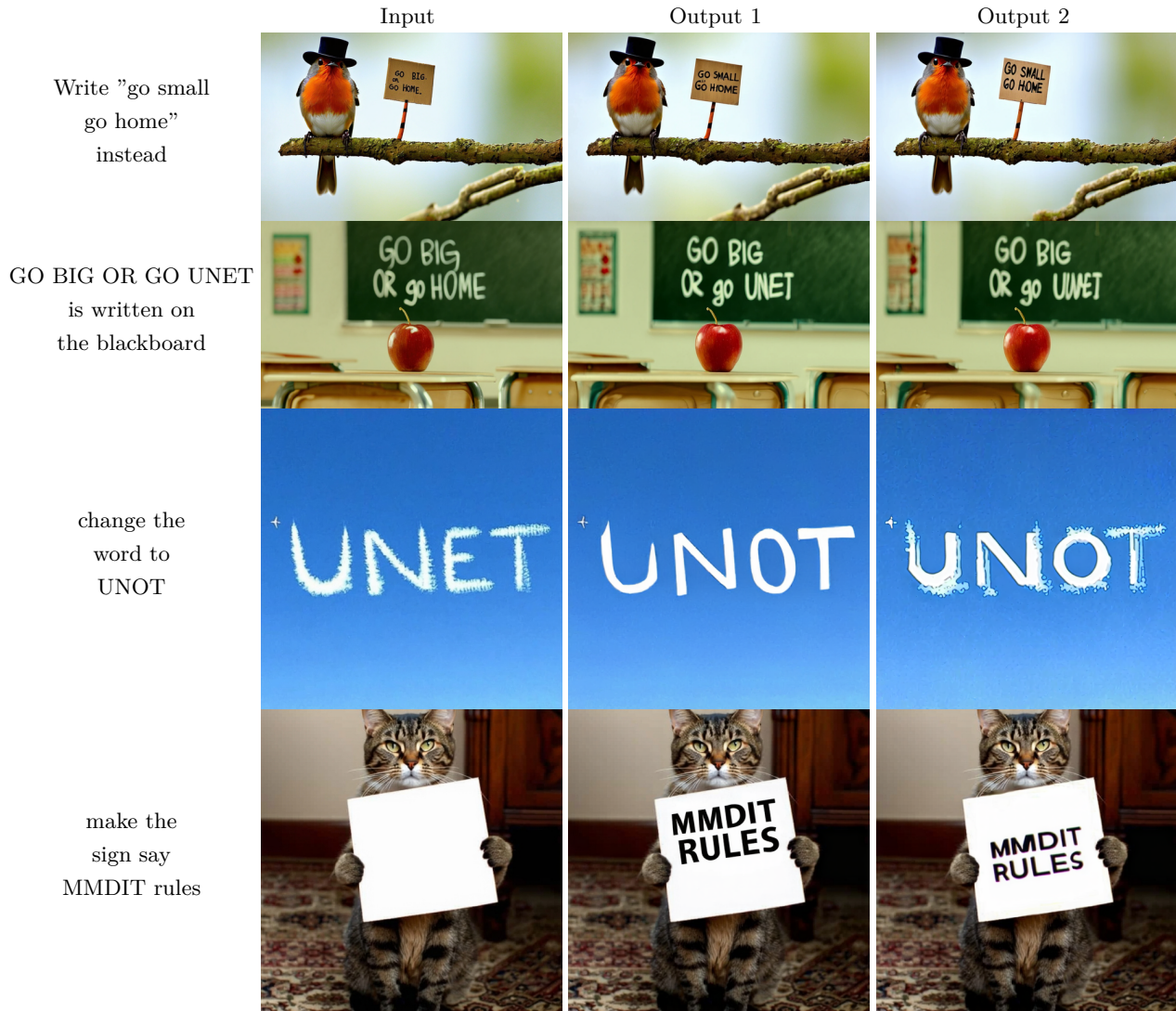


图 15. Zero Shot Text manipulation and insertion with the 2B Edit model
使用 2B Edit 模型进行零样本文本操作和插入

This comes with two main advantages: (i) The encoders do not need to be available on the GPU during training, lowering the required memory. (ii) The forward encoding pass is skipped during training, saving time and total needed compute after the first epoch, see Tab. 7.

我们的模型使用多个预训练冻结网络的输出作为输入（自编码器潜在和文本编码器表示）。由于这些输出在训练期间不变，我们为整个数据集预计算一次，带来两大优势。其一，编码器无需在 GPU 上可用，降低内存占用。其二，训练期间跳过前向编码过程，第一个 epoch 后可节省时间和总计算量，见 Tab. 7。

This approach has two disadvantages: First, random augmentation for each sample every epoch is not possible and we use square-center cropping during precomputation of image latents. For finetuning our model at higher resolutions, we specify a number of aspect ratio buckets, and resize and crop to the closest bucket first and then precompute in that aspect ratio. Second, the dense output of the text encoders is particularly large, creating additional storage cost and longer loading times during training (*c.f.* Tab. 7). We save the embeddings of the language models in half precision, as we do not observe a deterioration in performance in practice.

模型	内存 [GB]	前向传播 [ms]	存储 [kB]	增量 [%]
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

表 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).

预编码冻结输入网络的关键数据。内存是在 GPU 上加载模型所需的内存。前向传播时间 [ms] 是设备批量大小为 32 的每个样本前向传递的时间。存储是保存单个样本的大小。增量 [%] 是添加到 2B MMDiT 模型 (568ms/it) 循环中一个训练步骤增加的时间量。

该方法有两个缺点：首先，不能对每个 epoch 中的每个样本进行随机增强，我们在图像潜在的预计算期间使用平方中心裁剪。为了在更高分辨率下微调我们的模型，我们指定多个宽高比桶，先调整大小并裁剪到最近的桶，然后在该宽高比下进行预计算。其次，文本编码器的密集输出特别大，造成额外的存储成本和训练期间更长的加载时间（见 Tab. 7）。我们以半精度保存语言模型的嵌入，因为在实践中我们没有观察到性能的恶化。

E.2. 防止图像记忆化

在生成型图像模型的背景下，训练样本的记忆化可能导致许多问题 (Somepalli 等, 2023a; Carlini 等, 2023; Somepalli 等, 2023b)。为了避免我们训练的模型生成图像的逐字复制，我们仔细扫描训练数据集以查找重复的样本并将其删除。

Details on Deduplication In accordance with the methods outlined by Carlini 等 (2023) and Somepalli 等 (2023a), we opt for SSCD (Pizzi 等, 2022) as the backbone for the deduplication process. The SSCD algorithm is a state-of-the-art technique for detecting near-duplicate images at scale, and it generates high-quality image embeddings that can be used for clustering and other downstream tasks. We also decided to follow Nichol (2022) to decide on a number of clusters N . For our experiments, we use $N = 16,000$.

去重的细节 按照 Carlini 等 (2023) 和 Somepalli 等 (2023a) 提出的方法，我们选择 SSCD (Pizzi 等, 2022) 作为去重过程的骨干。SSCD 算法是大规模检测近似重复图像的最先进技术，它生成高质量的图像嵌入，可用于聚类和其他下游任务。我们也决定遵循 Nichol (2022) 来确定聚类数 N 。在我们的实验中，我们使用 $N = 16,000$ 。

We utilize `autofaiss` (2023) for clustering. `autofaiss` (2023) is a library that simplifies the process of using Faiss (Facebook AI Similarity Search) for large-scale clustering tasks. Specifically, leverage FAISS index factory¹ functionality to train a custom index with predefined number of centroids. This approach allows for efficient and accurate clustering of high-dimensional data, such as image embeddings.

我们利用 `autofaiss` (2023) 进行聚类。`autofaiss` (2023) 是一个库，简化了使用 Faiss (Facebook AI 相似度搜索) 进行大规模聚类任务的过程。具体地，我们利用 FAISS 索引工厂²功能训练具有预定义质心数的自定义索引。该方法允许对高维数据（如图像嵌入）进行高效精确的聚类。

Algorithm 1 details our deduplication approach. We ran an experiment to see how much data is removed by different SSCD threshold as shown in Figure 16b. Based on these results we selected four thresholds for the final run Figure 16a.

¹<https://github.com/facebookresearch/faiss/wiki/The-index-factory>

²<https://github.com/facebookresearch/faiss/wiki/The-index-factory>

Algorithm 1详细说明了我们的去重方法。我们进行了实验以查看不同 SSCD 阈值移除的数据量，如 Figure 16b所示。基于这些结果，我们为最终运行选择了四个阈值 Figure 16a。

E.3. 评估我们去重工作的效果

Carlini 等 (2023) 设计了一个两阶段的数据提取攻击，使用标准方法生成图像，并标记那些超过某些成员推理评分标准的图像。Carlini 等 (2023) 将他们的搜索偏向于重复的训练样本，因为这些样本比未重复的样本记忆化的可能性大数个数量级 (Somepalli 等, 2023a; Lee 等, 2021)。

To assess how well our SSCD-based deduplication works, we follow Carlini 等 (2023) to extract memorized samples from small, specifically for this purpose trained models and compare them before and after deduplication. Two main step of the mentioned procedure include: 1) Generate many examples using the diffusion model in the standard sampling manner and with the known prompts. 2) Perform membership inference to separate the model’s novel generations from those generations which are memorized training examples. Algorithm 2 shows the steps to find the memorized samples based on Carlini 等 (2023). Note that we run this techniques two times; one for SD-2.1 model with only exact dedup removal as baseline, and for a model with the SD2.1 architecture but trained on removed exact duplication and near-duplication using SSCD (Pizzi 等, 2022).

为了评估我们基于 SSCD 的去重效果如何，我们遵循Carlini 等 (2023) 从特别为此目的训练的小型模型中提取记忆化样本，并比较去重前后的结果。上述过程的两个主要步骤包括：1) 使用扩散模型以标准采样方式和已知提示生成许多样本。2) 执行成员推理以将模型的新颖生成与记忆化的训练样本分离。Algorithm 2基于 Carlini 等 (2023) 展示了查找记忆化样本的步骤。注意，我们运行了两次此技术；一次用于 SD-2.1 模型，以仅精确去重移除作为基线，一次用于具有 SD2.1 架构但在使用 SSCD (Pizzi 等, 2022) 移除精确重复和近似重复上训练的模型。

We select the 350,000 most-duplicated examples from the training dataset based on SSCD (Pizzi 等, 2022) with threshold of 0.5, and generate 500 candidate images for each text prompt to increase the likelihood of finding memorization. The intuition is that for diffusion models, with high probability $Gen(p; r_1) \approx_d Gen(p; r_2)$ for two different random initial seeds r_1, r_2 . On the other hand, if $Gen(p; r_1) \approx_d Gen(p; r_2)$ under some distance measure d , it is likely that these generated samples are memorized examples. To compute the distance measure d between two images, we use a modified Euclidean l_2 distance. In particular, we found that many generations were often spuriously similar according to l_2 distance (e.g., they all had gray backgrounds). We therefore instead divide each image into 16 non-overlapping 128×128 tiles and measure the maximum of the l_2 distance between any pair of image tiles between the two images.

我们从训练数据集中基于 SSCD (Pizzi 等, 2022) (阈值 0.5) 选出 350,000 个最重复的样本。每个提示生成 500 个候选图像以增加记忆化发现的概率。对于扩散模型，高概率下两个不同随机种子 r_1, r_2 生成的样本 $Gen(p; r_1) \approx_d Gen(p; r_2)$ 相似。若该相似性在距离度量 d 下成立，说明生成样本可能是训练样本的复制。我们用修改后的欧氏 l_2 距离计算两幅图像间的距离，但发现 l_2 距离常产生假相似（如都为灰色背景）。为解决这一问题，我们将每幅图分成 16 个不重叠的 128×128 方块，取任意方块对间 l_2 距离的最大值作为图像距离。Figure 17 shows the comparison between number of memorized samples, before and after using SSCD with the threshold of 0.5 to remove near-duplicated samples. Carlini 等 (2023) mark images within clique size of 10 as memorized samples. Here we also explore different sizes for cliques. For all clique thresholds, SSCD is able to significantly reduce the number of memorized samples. Specifically, when the clique size is 10, trained SD models on the deduplicated training samples cut off at SSCD= 0.5 show a $5\times$ reduction in potentially memorized examples.

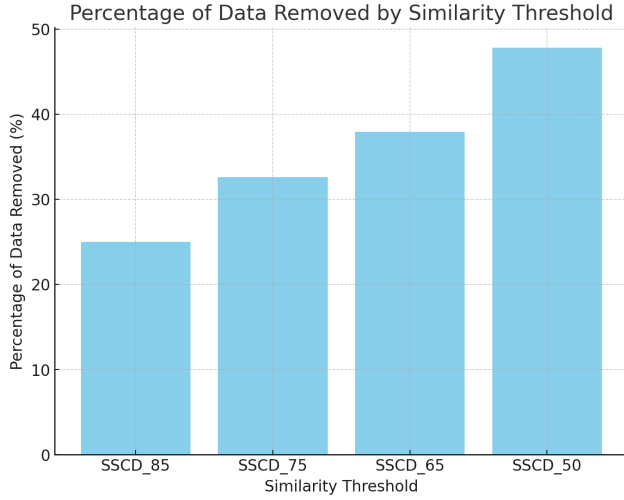
Figure 17展示了在使用阈值为 0.5 的 SSCD 移除近似重复样本前后记忆化样本数量的比较。Carlini 等 (2023) 将团大小为 10 内的图像标记为记忆化样本。我们也在探索了不同的团大小。对于所有团的阈值，SSCD 都能显著减少记忆化样本的数量。具体地，当团大小为 10 时，在阈值为 0.5 的 SSCD 处截断的去重训练样本上训练的 SD 模型显示了潜在记忆化例子的 5 倍减少。

算法 1 Finding Duplicate Items in a Cluster查找集群中的重复项

Require: *vecs* ——单个集群中的向量列表, *items* ——与 *vecs* 对应的条目 ID 列表, *index* ——用于集群内相似性搜索的 FAISS 索引, *thresh* ——判定重复的阈值

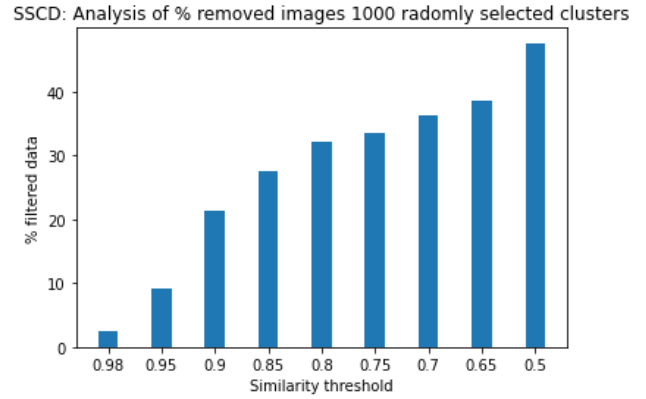
输出: *dups* ——重复条目 ID 的集合

```
1: dups  $\leftarrow$  new set()
2: for i  $\leftarrow$  0 to length(vecs) - 1 do
3:   qs  $\leftarrow$  vecs[i] {当前向量}
4:   qid  $\leftarrow$  items[i] {当前条目 ID}
5:   lims, D, I  $\leftarrow$  index.range_search(qs, thresh)
6:   if qid  $\in$  dups then
7:     continue
8:   end if
9:   start  $\leftarrow$  lims[0]
10:  end  $\leftarrow$  lims[1]
11:  duplicate_indices  $\leftarrow$  I[start : end]
12:  duplicate_ids  $\leftarrow$  new list()
13:  for j in duplicate_indices do
14:    if items[j]  $\neq$  qid then
15:      duplicate_ids.append(items[j])
16:    end if
17:  end for
18:  dups.update(duplicate_ids)
19: end for
20: 返回 dups {最终的重复 ID 集合}
```



(a) Final result of SS CD deduplication over the entire dataset

SS CD 去重在数据集上的最终结果



(b) Result of SS CD deduplication with various thresholds over 1000 random clusters

SS CD 去重在 1000 个随机簇上的各种阈值结果

图 16. Results of deduplicating our training datasets for various filtering thresholds. 我们训练数据集的去重结果，涵盖各种过滤阈值。

算法 2 Detecting Memorization in Generated Images

检测生成图像中的记忆化

Require: 提示集合 P 、每个提示的生成数量 N 、相似性阈值 $\epsilon = 0.15$ 、记忆化阈值 T

Ensure: 检测生成样本中被记忆化的图像

```

1: 将  $D$  初始化为重复度最高的样本集合
2: for 每个提示  $p \in P$  do
3:   for  $i = 1$  到  $N$  do
4:     用随机种子  $r_i$  生成图像  $\text{Gen}(p; r_i)$ 
5:   end for
6: end for
7: for 每一对生成图像  $x_i, x_j$  do
8:   if 距离  $d(x_i, x_j) < \epsilon$  then
9:     在图  $G$  中连接  $x_i$  和  $x_j$ 
10:  end if
11: end for
12: for  $G$  中的每个节点 do
13:   找出包含该节点的最大团
14:   if 团的大小  $\geq T$  then
15:     将团中的图像标记为被记忆化
16:   end if
17: end for

```

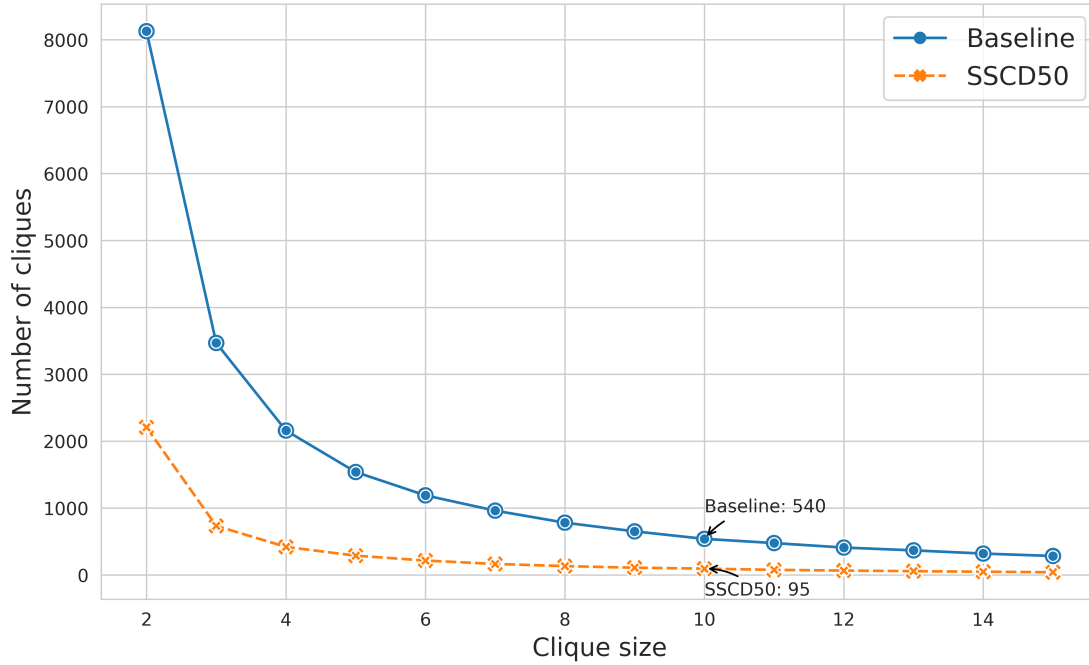



图 17. **SSCD-based deduplication prevents memorization.** To assess how well our SSCD-based deduplication works, we extract memorized samples from small, specifically for this purpose trained models and compare them before and after deduplication. We plot a comparison between number of memorized samples, before and after using SSCD with the threshold of 0.5 to remove near-duplicated samples. Carlini 等 (2023) mark images within clique size of 10 as memorized samples. Here we also explore different sizes for cliques. For all clique thresholds, SSCD is able to significantly reduce the number of memorized samples. Specifically, when the clique size is 10, models on the deduplicated training samples cut off at SSCD=0.5 show a $5\times$ reduction in potentially memorized examples.

基于 SSCD 的去重防止记忆化。为了评估我们基于 SSCD 的去重工作效果如何，我们从专门为此目的训练的小模型中提取记忆化样本，并比较去重前后的结果。我们绘制了记忆化样本数量的比较，对比了使用阈值为 0.5 的 SSCD 去除近重复样本前后的情况。Carlini 等 (2023) 将团大小为 10 的图像标记为记忆化样本。我们也探索了不同的团大小。对于所有团大小阈值，SSCD 能够显著减少记忆化样本的数量。具体来说，当团大小为 10 时，在 SSCD=0.5 处切割的去重训练样本上的模型显示出潜在记忆化示例数减少 $5\times$ 。