

# Rethinking Diffusion Model in High Dimension

郑镇鑫

blair.star@163.com

## Abstract

维数灾难是统计概率模型难以逃脱的问题，但扩散模型似乎打破了这一限制，在高维数据生成上取得良好的结果。扩散模型假设模型能学习到隐藏概率分布的统计性质，从而能从隐藏概率分布进行采样，生成真实的样本，但工作原理真的是这样子吗？为了解答这个问题，本文对扩散的目标函数和推断方法展开仔细的分析，发现了几个重要的结论，有助于回答上面的问题：1) 在高维稀疏场景下，目标函数的拟合对象从“多个样本的加权和”退化成“单个样本”；2) 当前主流的推断方法都可用一个简单的框架表示，无需涉及 Markov chain 和 SDE 等统计概念。3) 以简单框架为指导，可找到更高效的推断方法。

## 1 Introduction

扩散模型在高维数据生成场景下呈现出突出的竞争力，特别是在图像生成领域 [Rom+22]。扩散模型不仅仅拥有突出的效果，还具有多种优雅的数学形式。[SD+15] 首先提出了扩散模型的方法，此方法通过马尔可夫链，把复杂的数据分布转化为简单的正态分布，然后学习转换过程对应的后验概率分布。[HJA20] 对扩散模型的目标函数的形式作了改进优化，提出了 Denoised DPM。[Son+20] 对扩散模型的加噪方式进行泛化，从离散泛化到连续，并抽象为统计微分方程。[Lip+22; LGL22] 从流匹配的角度提出了一种新的优化思路，让模型直接学习概率流的速度场。

上述三种模型均假设模型能学习到数据分布的统计性质：对于 Markov Chain 的形式，假设其能够学习得到后验概率分布；对于统计微分方程的形式，假设其能够学习到边际分布的 Score；对于流匹配，假设能够学习到速度场。但此观点与过往的认识有一些矛盾，过往一般认为，在高维稀疏场景中，机器学习模型是不能够有效学习到隐藏的复杂概率分布的。于是，有必要重新思考一下，扩散模型是否能够学习到复杂的概率分布，是否是以假定的方式发挥着作用。

针对这些疑问，本文将提供几个方面的分析，为更进一步地理解高维扩散模型提供一些贡献。

首先，分析稀疏性对扩散模型目标函数的影响，当数据集足够稀疏时，目标函数退化成另外一种形式，由“加权和”退化成“单一样本”。并且，以新的目标函数形式为基础，提出一种新的目标函数的理解方式。

其次，提出一个新的推断框架。新框架不仅与新的目标函数理解方式相匹配，还统一了大部分主流的推断方法，包括 DDPM 祖先采样，DDIM[SME20]，Euler、dpm-solver[Lu+22a]、dpm-solver++[Lu+22b] 及 DEIS[ZC22] 等。

最后，以新框架为基础，设计了新的推断方法，并通过实验证明其比当前主流的推断方法更为高效。

## 2 Background

给定随机变量  $X_0$  的一批采样数据  $X_0^0, X_0^1, \dots, X_0^N$ ，扩散模型以不同的比例混合数据与随机噪声，从而形成一系列新变量  $X_1, X_2, \dots, X_T$ ，新变量的信噪比（数据与噪声比例）逐渐减小，到最后一个变量时，几乎包含随机噪声。

对于原始的扩散模型及 VP SDE，其以如下的方式进行混合：

$$X_t = \sqrt{\bar{\alpha}_t} \cdot X_0 + \sqrt{1 - \bar{\alpha}_t} \cdot \epsilon \quad (1)$$

其中， $\bar{\alpha}_t$  从 1 逐渐减小至 0， $t$  取离散值，从 1 至  $T$ 。

对于流匹配，其以如下方式进行混合：

$$X_t = (1 - \sigma_t) \cdot X_0 + \sigma_t \cdot \epsilon \quad (2)$$

其中， $\sigma_t$  也从 1 逐渐减小至 0，在实践中，常设  $\sigma_t = t$ 。 $t$  取连续值， $t \in [0, 1]$ 。

## 2.1 Markov Chain-based Diffusion Models

对于基于 Markov Chain 的扩散模型，其核心在于学习条件后验概率  $p(x_{t-1}|x_t)$ ，由于后验概率近似高斯函数，并且其方差相对固定，所以可只学习条件后验概率的均值  $E_{p(x_{t-1}|x_t)}(x_{t-1})$ 。根据 Total Law of Expectation[Ros10]，均值可表示成另一种形式：

$$E_{p(x_{t-1}|x_t)}(x_{t-1}) = \int p(x_0|x_t) E_{p(x_{t-1}|x_0, x_t)}(x_{t-1}) dx_0 \quad (3)$$

由 [HJA20] 的公式 (7) 可知， $p(x_{t-1}|x_0, x_t)$  的均值可表示成  $x_0$  和  $x_t$  的线性组合形式，即

$$E_{p(x_{t-1}|x_0, x_t)}(x_{t-1}) = \underbrace{\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} \cdot x_0}_{const=C_0} + \underbrace{\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} \cdot x_t}_{const=C_t} \quad (4)$$

于是， $p(x_{t-1}|x_t)$  的均值可进一步表示成

$$E_{p(x_{t-1}|x_t)}(x_{t-1}) = \int p(x_0|x_t) (C_0 \cdot x_0 + C_t \cdot x_t) dx_0 \quad (5)$$

$$= C_0 \int p(x_0|X_t) x_0 dx_0 + C_t \cdot x_t \int p(x_0|x_t) dx_0 \quad (6)$$

$$= C_0 \int p(x_0|x_t) x_0 dx_0 + C_t \cdot x_t \quad (7)$$

所以，基于 Markov Chain 扩散模型的学习目标可认为是学习条件概率分布  $p(x_0|x_t)$  的均值 ( $\int p(x_0|x_t) x_0 dx_0$ )，即

$$\min_{\theta} \int p(x_t) \left\| f_{\theta}(x_t) - \int p(x_0|x_t) x_0 dx_0 \right\|^2 dx_t \quad (8)$$

其中， $f_{\theta}(x_t)$  为可学习的神经网络函数，输入为  $x_t$ 。

## 2.2 Score-based Diffusion Model

基于 Score 的扩散模型，其核心在于学习边际分布  $p(x_t)$  的 Score( $\frac{\partial \log p(x_t)}{\partial x_t}$ )。与上述原始扩散模型类似，通过引入另一个变量  $X_0$ ，Score 可表示另一种形式：

$$\frac{\partial \log p(x_t)}{\partial x_t} = \int p(x_0|x_t) \frac{\partial \log p(x_t|x_0)}{\partial x_t} dx_0 \quad (9)$$

上述关系的证明可看附录 A.2。

由于  $p(x_t|x_0) \sim \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, \sqrt{1-\bar{\alpha}_t})$ ，所以  $p(x_t|x_0)$  的 Score 可表示成

$$\frac{\partial \log p(x_t|x_0)}{\partial x_t} = -\frac{x_t - \sqrt{\bar{\alpha}_t}x_0}{1-\bar{\alpha}_t} = \underbrace{\frac{\sqrt{\bar{\alpha}_t}}{1-\bar{\alpha}_t} \cdot x_0}_{const=S_0} + \underbrace{\frac{-1}{1-\bar{\alpha}_t} \cdot x_t}_{const=S_t} \quad (10)$$

于是， $p(x_t)$  的 score 可表示成

$$\frac{\partial \log p(x_t)}{\partial x_t} = \int p(x_0|x_t) (S_0 \cdot x_0 + S_t \cdot X_1) dx_0 \quad (11)$$

$$= S_0 \int p(x_0|x_t) x_0 dx_0 + S_t \cdot X_1 \quad (12)$$

因此，基于 score 的扩散模型的学习目标也可认为是学习条件概率分布  $p(x_0|x_t)$  的均值 ( $\int p(x_0|x_t) x_0 dx_0$ )。

### 2.3 Flow Matching-based Diffusion

基于 Flow Matching 的扩散模型的核心在于学习概率流的速度场 (velocity field)。根据 [Lip+22] 中的 Theorem 1，速度场  $u(x_t)$  可表示成条件速度场  $u(x_t|x_0)$  的加权和，即

$$u(x_t) = \int p(x_0|x_t) u(x_t|x_0) dx_0 \quad (13)$$

而由式2可知，条件速度场  $u(x_t|x_0)$

$$u(x_t|x_0) \triangleq \frac{dx_t}{dt} = \epsilon - x_0 \quad (14)$$

于是，速度场  $u(x_t)$  可表示成

$$u(x_t) = \int p(x_0|x_t) (\epsilon - x_0) dx_0 = \epsilon - \int p(x_0|x_t) x_0 dx_0 \quad (15)$$

所以，基于 Flow Matching 的扩散模型的学习目标也可认为是学习条件概率分布  $p(x_0|x_t)$  的均值 ( $\int p(x_0|x_t) x_0 dx_0$ )。

### 2.4 Predicting Posterior Mean is Equivalent to Predicting $X_0$

拟合条件后验概率  $p(x_0|x_t)$  的均值与预测  $X_0$  是等价的，即

$$\min_{\theta} \int p(x_t) \left\| f_{\theta}(x_t) - \int p(x_0|x_t) x_0 dx_0 \right\|^2 dx_t \iff \min_{\theta} \iint p(x_0, x_t) \|f_{\theta}(x_t) - x_0\|^2 dx_0 dx_t \quad (16)$$

具体证明可见附录A.1。

## 3 Impact of Sparsity on the Objective Function

由 [section:background] 节可知，扩散模型的核心在于学习拟合各个  $x_t$  对应的后验概率分布  $p(x_0|x_t)$  的均值，但是由于  $x_t$  的取值有无限多，所以并不能无限遍历。实际的作法是根据  $p(x_t)$  的概率大小采样  $x_t$ ，概率大的优先拟合，但由于  $p(x_t)$  是未知的，不能直接采样，所以是通过祖先采样的方式间接采样。具体采样方法如下：从  $p(x_0)$  采样得到  $X_0$ ，然后根据  $p(x_t|x_0)$  采样得到  $X_t$ ， $(X_0, X_t)$  配对服从  $p(x_0, x_t)$ ，单独的  $X_t$  服从  $p(x_t)$ 。

接下来，我们分析优化过程中所遇到的  $p(x_t|x_0)$  的均值的特点。为了分析均值的具体特点，首先介绍  $p(x_0|x_t)$  分布的形式。

### 3.1 Form of Posterior $p(x_0|x_t)$

为了方便，使用统一的形式表示式(1)和式(2)两种混合方式：

$$x_t = c_0 \cdot x_0 + c_1 \cdot \epsilon \quad (17)$$

当  $c_0^2 + c_1^2 = 1$  时，属于基于 Markov Chain 的扩散模型和基于 Score 扩散模型的混合方式；当  $c_0 + c_1 = 1$  时，属于 Flow Matching 的混合方式。在此表示下， $p(x_t|x_0) \sim \mathcal{N}(x_t; c_0 x_0, c_1^2)$ 。

由附录A.3的分析可知，后验概率分布  $p(x_0|x_t)$  有如下的形式：

$$p(x_0|x_t) = \text{Normalize} \left( \exp \frac{-(x_0 - \mu)^2}{2\sigma^2} p(x_0) \right) \quad \text{where } \mu = \frac{x_t}{c_0} \quad \sigma = \frac{c_1}{c_0} \quad (18)$$

在实际优化中， $p(x_0)$  是隐藏的数据分布，是未知的，是不能够直接采样的，只能从已有的样本  $\{X_0^0, X_0^1, \dots, X_0^N\}$  中随机挑选 ( $X_0^i \sim p(x_0)$ )，挑选过程可认为是从下面的混合 Dirac delta distribution 中采样：

$$p(x_0|x_t) = \frac{1}{N} \sum_{i=0}^N \delta(x_0 - X_0^i) \quad (19)$$

把上式代入(18), 可得

$$p(x_0|x_t) = \text{Normalize} \left( \exp \frac{-(x_0 - \mu)^2}{2\sigma^2} \sum_{i=0}^N \delta(x_0 - X_0^i) \right) \quad (20)$$

$$= \frac{1}{Z_c} \exp \frac{-(x_0 - \mu)^2}{2\sigma^2} \sum_{i=0}^N \delta(x_0 - X_0^i) \quad (21)$$

$$\text{where } \mu = \frac{x_t}{c_0} \quad \sigma = \frac{c_1}{c_0} \quad Z_c = \sum_{i=0}^N \exp \frac{-(X_0^i - \mu)^2}{2\sigma^2} \quad (22)$$

可以看出, 当  $p(x_0)$  是离散分布时,  $p(x_0|x_t)$  也是离散分布, 并且每个离散值  $X_0^i$  的概率大小与  $X_0^i$  跟  $\mu$  的距离反相关。

### 3.2 Weighted Sum Degradation Phenomenon

上面已经介绍了  $p(x_0|x_t)$  的具体形式, 于是, 可进一步分析  $p(x_0|x_t)$  的均值的特点。根据均值的定义,  $p(x_0|x_t)$  的均值可表示为:

$$\int x_0 p(x_0|x_t) dx_t = \frac{1}{Z_c} \sum_{i=0}^N X_0^i \exp \frac{-(X_0^i - \mu)^2}{2\sigma^2} \quad (23)$$

$$\text{where } \mu = \frac{x_t}{c_0} \quad \sigma = \frac{c_1}{c_0} \quad Z_c = \sum_{i=0}^N \exp \frac{-(X_0^i - \mu)^2}{2\sigma^2} \quad (24)$$

可以看出,  $p(x_0|x_t)$  的均值是所有的  $X_0$  样本的加权和, 权重大小跟  $X_0^i$  与  $\mu$  的距离的反比。

此加权和有一个特点, 当只有一个样本与  $\mu$  特别近, 其它样本与  $\mu$  比较远时, 加权和退化成单个样本。在数据比较稀疏时, 这种现象将非常明显。

图1给出了一个简单的示例, 蓝色的点是数据样本, 绿色圆代表所加高斯噪声的标准差, 可以看出, 当样本比较稀疏, 高斯标准差半径相对较小时, 大量  $X_t$  样本将只分布蓝色点附近的局部区域内, 而对应的后验概率  $p(x_0|x_t)$  将收缩成 Dirac Delta distribution, 也就是说  $p(x_0|x_t)$  将只有一个非零值, 位于附近惟一的蓝色点。因此, 对应的  $p(x_0|x_t)$  的均值将从加权和退化成单个样本。在本文中, 将此现象称之为“加权和退化”。

“加权和退化”将直接影响模型是否能够学习到隐藏的数据分布。

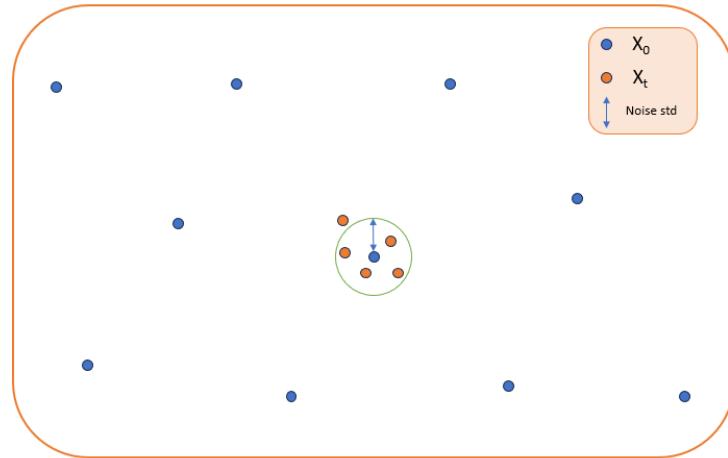


Figure 1: Impact of data sparsity on posterior probability distribution

下面将分析两个生成模型常用数据集的“加权和退化”的程度: Imagenet-256 和 Imagenet-512。两个数据集的维度相对较高, 像素维度分别 196608 和 786432, 使用 VAE[KW+13; Rom+22] 压缩后, 隐空间的维度也比较高, 分别达到 4096 和 16480。因为压缩的做法更常用, 所以下面只分析压缩的情况。

为了定量描述“加权和退化”的程度, 我们作如下的定义: 从  $p(x_0|x_t)$  采样一个样本  $X_0$ , 根据  $t$  值混合相应幅度的随机噪声, 得到  $X_t$ , 此方式得到的  $X_t \sim p(x_t)$ , 然后分析概率分布  $p(x_0|x_t = X_t)$  的特点, 如果

Table 1: ImageNet256 统计结果

merging\time	200	300	400	500	600	700	800	900
vp	1.00/1.00	1.00/1.00	1.00/0.98	0.91/0.57	0.41/0.01	0.02/0.00	0.00/0.00	0.00/0.00
flow	1.00/1.00	1.00/1.00	1.00/1.00	1.00/1.00	1.00/0.95	0.97/0.69	0.76/0.15	0.09/0.00

Table 2: ImageNet512 统计结果

merging\time	200	300	400	500	600	700	800	900
vp	1.00/1.00	1.00/1.00	1.00/0.98	0.98/0.57	0.87/0.08	0.50/0.00	0.03/0.00	0.00/0.00
flow	1.00/1.00	1.00/1.00	1.00/1.00	1.00/1.00	1.00/0.94	0.99/0.67	0.95/0.20	0.71/0.01

存在一个  $x_0 = X'_0$ , 其对应  $p(x_0 = X'_0 | x_t = X_t)$  的概率大于 0.9, 那么认为这个概率分布近似为 Dirac Delta distribution, 出现“加权和退化”现象; 如果  $X'_0 = X_0$ , 即退化成第一步采样的结果样本  $X_0$ , 则称之为“加权和退化至  $X_0$ ”。可以看出, 这里使用的采样方式与实际训练中所使用的采样方式一样的, 这样能够更好地判断优化过程中所拟合的  $E_{p(x_0|x_t=X_t)}(x_0)$  是否会退化成  $X_0$ 。

由上面可知, 除了数据的稀疏程度外, 噪声标准差大小也会影响“加权和退化”, 所以, 对不同的  $t$  值进行单独的统计。Imagenet 有 1000 个类, 每个类大概采样 1000 次, 于是, 每个  $t$  大概采样 1000 万次, 分别统计“加权和退化”和“加权和退化至  $X_0$ ”的比例。我们分析两种不同的噪声混合模式: 原始 DDPM 的噪声混合模式 (式(1)) 和 Flow Matching 的混合模式 (式(2))。

表格3.2给出了 imagenet-256 的具体统计结果, 表格3.2给出了 imagenet-512 的具体统计结果。可以看出几个明显的规律:

- 随着  $t$  的减小, 加权和退化现象越来越明显。
- Flow Matching 比 VP 更多出现退化现象。
- 维度越高, 出现退化的比例越大。

另外, 我们可以看到, 不管是 VP 还是 Flow Matching, 退化现象在这两个数据集中相当严重, 当  $t < 600$  时, 绝大多数均值都退化成单个样本。实际上, 在优化过程中, 由于采样次数的限制, 每个  $p(x_0|x_t = X_t)$  都不能充分采样, 所以实际的退化比例应比统计结果更高。

由于高维的数据分布一般都比较复杂, 对应的  $p(x_0|x_t = X_t)$  也应比较复杂, 当出现加权和退化时, 相当于使用单一个样本作为均值估计, 此估计一般会有比较大的误差。如果我们不能够提供一个较准确的拟合目标, 可推断模型也不可能学习到理想准确的目标。因此, 我们有必要重新思考一下, 扩散模型是否能够学习到隐藏的概率分布, 是否是按照概率流动的方式进行着工作。

### 3.3 A Simple Way to Understand the Objective Function

由上一小节可知, 高维空间的扩散模型的拟合目标存在大量的“加权和退货现象”, 并且相当大部分是退化至祖先采样的第一步的结果 (即  $X_0$ )。另外, 我们知道, 第二步采样是一个混合噪声过程, 于是可以用简单的方式理解高维扩散模型的目标函数: 从混合噪声后的信号中预测原始信号。

如果从频谱的角度进行分析, 可进一步理解其原理 [Die24]。

在频率域, 普通图像信号的频谱是非均匀分布的, 低频分量占据大部分, 高频分量占据极少部分, 具体可看图2(a) 和 (b), 下半部分是频谱图, 中心是低频部分, 外围是高频部分, 可以看到, 中心较亮, 外围较暗; 高斯噪声信号与普通信号不同, 频谱是均匀分布的, 也就是说各频率分量均相同, 具体可看图2(c) 和 (d), 频谱图为一个均匀亮度的平面。

于是, 如果把一个图像信号与随机噪声混合, 高频分量的信噪比总是小于低频分量的信噪比, 随着, 噪声幅度的增大, 高频分量将最早被淹没, 然后逐渐淹没至低频分量。通过图3可帮助理解此过程。

当神经网络从加噪信号中优化预测原始图像信号时, 由于各频率分量的信噪比不同, 被拟合的优先顺序将各不相同。暂且把频率分量分为两类, 一类是淹没的, 另一类是未被淹没的。对于没被淹没的部分, 神经网络将

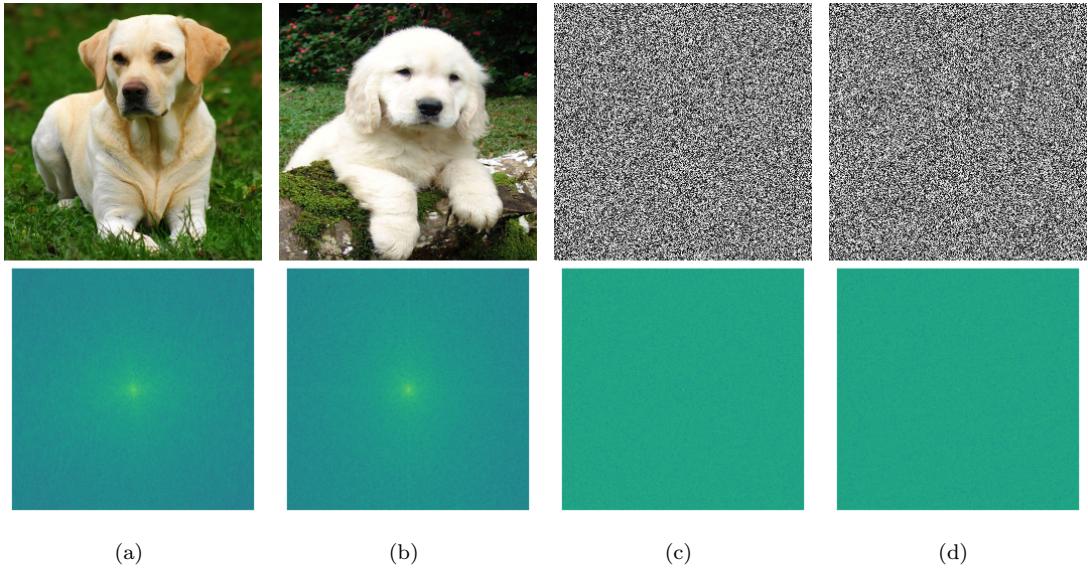


Figure 2: (a)(b) normal image and its frequency spectral (c)(d) gaussian noise and its frequency spectral

能轻松预测；对于被淹没的部分，神经网络将优先预测较低频率分量，原因有两点：较低频分量有更高的信噪比，在更容易被预测；低频分量有更大的幅度，相当于在欧氏距离损失函数中有更大的权重。

于是，目标函数可理解为“过滤较高频分量-补全被过滤的频率分量”，如图4所示。当  $t$  比较大时，噪声淹没比较多的频率分量，不仅高频分量，也包括部分低频分量，此时，模型优先预测低频分量。当  $t$  比较小，噪声只淹没高频分量，此时，模型直接预测高频分量。

在推断过程经常看到，早期预测的  $X_0$  只包含着轮廓信息 ( $t$  比较大)，后期预测的  $X_0$  包含着更多细节信息 ( $t$  比较小)，这种现象印证了上面的分析。

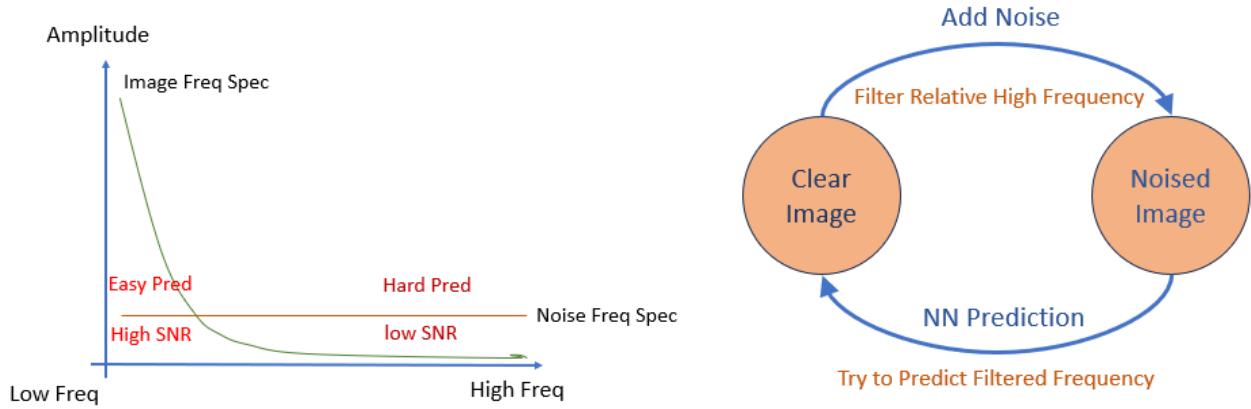


Figure 3: Image and noise frequency spectrum

Figure 4: New perspective of training object function

## 4 A Unified Inference Framework-Natural Inference

我们知道，目前扩散模型的推断方法都是以较严格的数学方法设计的，包括基于 Markov-Chain 的祖先采样、基于统计微分方程的逆过程求解等等。但是，在上一节中，我们知道扩散模型的目标函数能以一种比较简单的方式理解-从加噪图像中预测原始图像，那么，基于训练-测试-相匹配的原则，我们自然会有疑问，目前的推断方法是否也可以以一种类似的方式理解呢？

答案是肯定的，下面我们将揭示，目前的大多数推断方法都可统一成一种基于预测  $x_0$  的简单框架，包括祖先采样、DDIM、Euler、DPMsolver、DPMsolver++、DEIS 以及 Flow Matching 的求解方法等等。

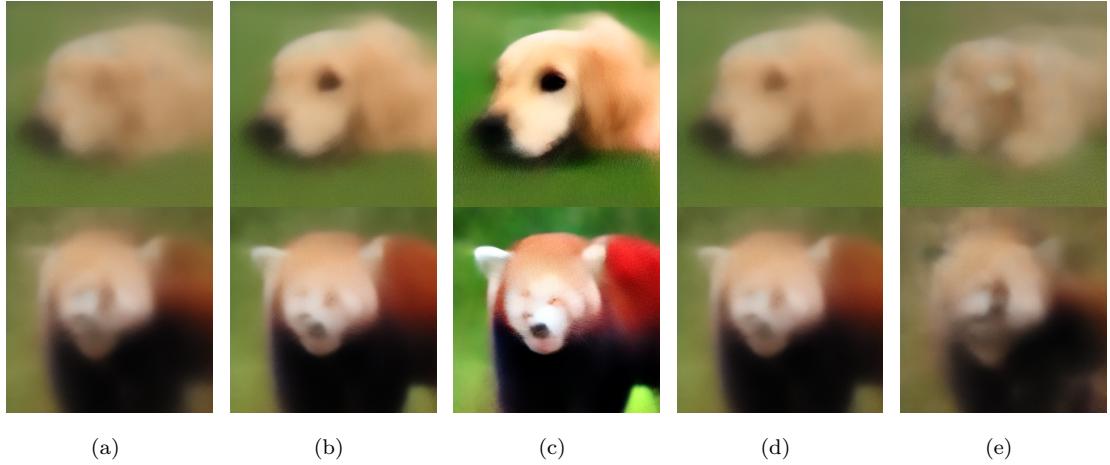


Figure 5: (a) model output on  $t=540$  (b) model output on  $t=500$  (c) fore Self Guidance (d) mid Self Guidance (e) back Self Guidance

在介绍新框架之前，先介绍新框架所包含的一类关键操作。

#### 4.1 Self Guidance

首先，仿照 Classifier Free Guidance[HS22] 引入一个新操作，叫 Self Guidance。Classifier Free Guidance 原理可总结如下：

$$I_{out} = I_{bad} + \lambda \cdot (I_{good} - I_{bad}) \quad (25)$$

其中， $I_{bad}$  是较差模型的输出， $I_{good}$  是较好模型的输出，两个模型共享同样的输入， $\lambda$  控制 Guidance 的程度。

实际上，Classifier Free Guidance 与传统图像处理中的 Unsharp Masking 锐化 [GW17] 有些类似，在 Unsharp masking 锐化算法中， $I_{good}$  是原始的图像， $I_{bad}$  是经过高斯模糊后图像， $(I_{good} - I_{bad})$  得到图像的边缘信息，把边缘信息加入到原始图像  $I_{good}$  中，将得到一个边缘更锐利的图像。因此，Classifier Free Guidance 也可以认为是一种图像增强操作。

扩散模型的推断过程中，会得到一系列预测的  $x_0$ ，起初  $x_0$  的质量较差，越往后  $x_0$  的质量越高。如果用较早的  $x_0$  作为  $I_{bad}$ ，用较后的  $x_0$  作为  $I_{good}$ ，那么，在本文中，把这个操作叫作 Self Guidance，因为  $I_{bad}$  和  $I_{good}$  都是本身的模型输出，不需引入额外的模型。

依据  $\lambda$  的取值，把 Self Guidance 进一步地分类。

- 当  $\lambda > 1$  时，称之为 **Fore Self Guidance**，此时，输出有提升质量的作用。
- 当  $0 < \lambda < 1$  时，称之为 **Middle Self Guidance**，此时，输出在  $I_{bad}$  和  $I_{good}$  之间线性插值，比  $I_{good}$  的质量差，比  $I_{bad}$  质量好。
- 当  $\lambda < 0$  时，称之为 **Back Self Guidance**，此时，输出在不仅比  $I_{good}$  的质量更差，也比  $I_{bad}$  质量更差。

图5给出了两个例子，(a) 是较早的模型输出 ( $t=540$ )，质量较差 ( $I_{bad}$ )，(b) 是较晚的模型输出 ( $t=500$ )，质量较好 ( $I_{good}$ )，(c) 是 Fore Self Guidance 的结果，(d) 是 Mid Self Guidance 的结果，(e) 是 Back Self Guidance 的结果。

式(25)可进一步写成

$$I_{out} = \lambda \cdot I_{good} + (1 - \lambda) \cdot I_{bad} \quad (26)$$

$$= C_{good} \cdot I_{good} + C_{bad} \cdot I_{bad} \quad (27)$$

$$\text{where } C_{good}, C_{bad} \in \text{real} \quad C_{good} + C_{bad} = 1 \quad (28)$$

上面表明,  $I_{bad}$  和  $I_{good}$  的系数可以取任意值, 但两者  $I_{bad}$  和  $I_{good}$  之和必等于 1。对于 **Fore Self Guidance**,  $C_{good} > 0$ ,  $C_{bad} < 0$ ; 对于 **Middle Self Guidance**,  $C_{good} > 0$ ,  $C_{bad} > 0$ ; 对于 **Back Self Guidance**,  $C_{good} < 0$ ,  $C_{bad} > 0$ 。

于是, 存在下面的结论:

对于任意两个  $I_{bad}$  和  $I_{good}$  的线性和, 都可表示成 Self Guidance 与一个比例系数, 即

$$I_{out} = a \cdot I_{good} + b \cdot I_{bad} = (a + b) \cdot \left( \frac{a}{a + b} \cdot I_{good} + \frac{b}{a + b} \cdot I_{bad} \right) \quad (29)$$

可以看出, 括号里操作是一个 Self Guidance 操作, 因为两个系数之和为 1。

## 4.2 Natural Inference

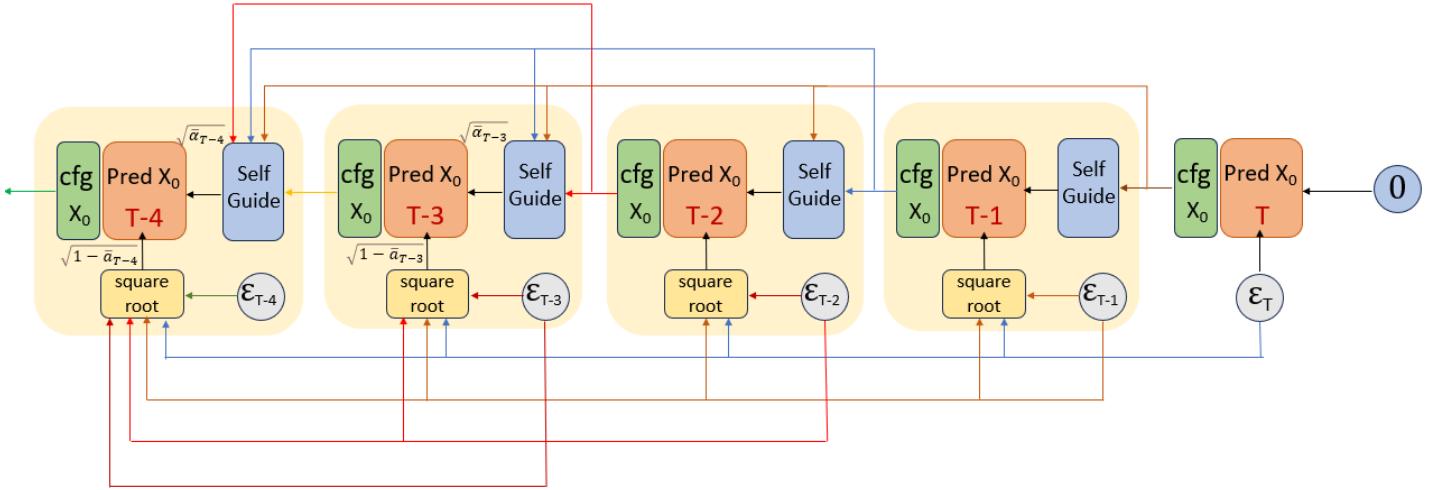


Figure 6: A new inference framework - Natural Inference

新推断框架的思路如公式(30)及图6所示, 包含  $T$  个预测  $X_0$  的操作, 从最大的  $t$  对应的模型开始执行预测, 然后顺序过渡到较小的  $t$ , 每个预测模型包含两部分的输入, 包括图像信号的输入和噪声信号的输入, 两者比例与训练阶段的混合比例保持相同。图像信号由过往的模型的输出的线性加权和构成; 噪声信号由过往的噪声和当前新加的噪声线性加权和构成。由4.1可知, 线性加权和可理解为多个 Self Guidance 操作的复合。由于各个噪声是相互独立的, 所以此处采用平方和的根计算总噪声的幅度。

由于当前模型的输入图像信号依赖于过往模型的输出图像信号, 所以此推断框架具有**自回归**的形式, 只不过其对过往图像信号的加工处理较为简单, 只是一个简单的加权和。

$$\begin{aligned}
y_T &= \mathbf{f}_T(\varepsilon_T) \\
y_{T-1} &= \mathbf{f}_{T-1}(N_{T-1} + S_{T-1}) \\
\text{where } N_{T-1} &= b_{T-1}^{T-1} \varepsilon_{T-1} + b_{T-1}^T \varepsilon_T \quad S_{T-1} = c_{T-1}^T y_T \\
\sqrt{(b_{T-1}^{T-1})^2 + (b_{T-1}^T)^2} &= \sqrt{1 - \bar{\alpha}_{T-1}} \quad c_{T-1}^T = \sqrt{\bar{\alpha}_{T-1}} \\
&\dots \\
y_t &= \mathbf{f}_t(N_t + S_t) \tag{30} \\
\text{where } N_t &= \sum_{i=t}^T b_i^i \varepsilon_i \quad S_t = \sum_{i=t+1}^T c_i^i y_i \\
\sqrt{\sum_{i=t}^T (b_i^i)^2} &= \sqrt{1 - \bar{\alpha}_t} \quad \sum_{i=t+1}^T c_i^i = \sqrt{\bar{\alpha}_t} \\
&\dots
\end{aligned}$$

$\begin{array}{cccc} y_T & y_{T-1} & y_{T-2} & y_{T-3} \\ \left( \begin{array}{cccc} c_{T-1}^T & 0 & 0 & 0 \\ c_{T-2}^T & c_{T-2}^{T-1} & 0 & 0 \\ c_{T-3}^T & c_{T-3}^{T-1} & c_{T-3}^{T-2} & 0 \\ c_{T-4}^T & c_{T-4}^{T-1} & c_{T-4}^{T-2} & c_{T-4}^{T-3} \end{array} \right) & \left( \begin{array}{ccccc} \varepsilon_T & \varepsilon_{T-1} & \varepsilon_{T-2} & \varepsilon_{T-3} & \varepsilon_{T-4} \\ b_{T-1}^T & b_{T-1}^{T-1} & 0 & 0 & 0 \\ b_{T-2}^T & b_{T-2}^{T-1} & b_{T-2}^{T-2} & 0 & 0 \\ b_{T-3}^T & b_{T-3}^{T-1} & b_{T-3}^{T-2} & b_{T-3}^{T-3} & 0 \\ b_{T-4}^T & b_{T-4}^{T-1} & b_{T-4}^{T-2} & b_{T-4}^{T-3} & b_{T-4}^{T-4} \end{array} \right) \\ \text{Signal Coefficient Matrix} & \text{Noise Coefficient Matrix} \end{array}$

Figure 7: Coefficient matrix on the natural inference framework

其中， $\mathbf{f}_t$  是第  $t$  步预测  $x_0$  的模型函数， $y_t$  是  $\mathbf{f}_t$  模型函数的输出，即第  $t$  步所预测的  $x_0$ ， $\varepsilon_t$  是相互独立的标准高斯噪声， $b_t^i$  和  $c_t^i$  是常量系数。独立高斯噪声之和也是高斯噪声，其标准差等于原各个标准差的平方和之根。

在本文中，把  $\sqrt{\bar{\alpha}_t}$  称为信号边际系数， $\sqrt{1 - \bar{\alpha}_t}$  称之为边际噪声系数， $\sum_{i=t+1}^T c_i^i$  称之为等效边际信号系数， $\sqrt{\sum_{i=t}^T (b_i^i)^2}$  称之为等效边际噪声系数。为了方便展示所有的  $c_t^i$  和  $b_t^i$ ，本文把所有的系数组织成矩阵的形式，如图7所示，这个矩阵称为系数矩阵。可以看到，由于自回归的性质，信号系数矩阵呈现出下三角的形式。

### 4.3 Represent DDPM Ancestral Sampling with Natural Inference Framework

接下来，我们将揭示，DDPM 的祖先采样算法，可等价转化成 Natural Inference 框架的形式。祖先采样算法的迭代方式如下：

$$\begin{aligned}
y_t &= f_t(x_t) \\
x_{t-1} &= d_{t-1} \cdot x_t + e_{t-1} \cdot y_t + g_{t-1} \cdot \varepsilon_{t-1} \tag{31} \\
\text{where } d_{t-1} &= \frac{\sqrt{\bar{\alpha}_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} \quad e_{t-1} = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t} \quad g_{t-1} = \sqrt{\frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t}}\beta_t
\end{aligned}$$

其中， $f_t$  是第  $t$  步的模型函数，在这里，假设模型是预测  $x_0$  的形式，其它预测形式的模型，都可转化成预测  $x_0$  的形式。 $y_t$  是  $\mathbf{f}_t$  的输出，即第  $t$  步所预测的  $x_0$ 。

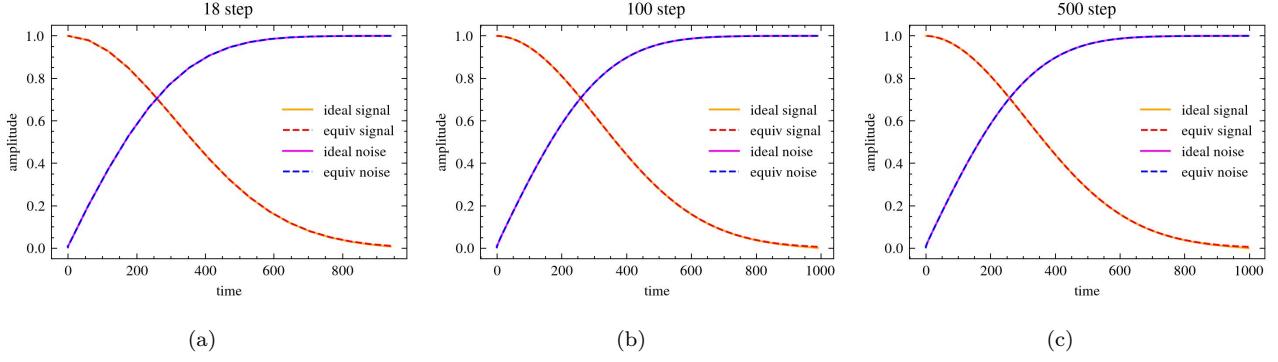


Figure 8: DDPM equivalent marginal coefficients and ideal marginal coefficients (a) 18 step (b) 100 step (c) 500 step

根据上面的迭代算法,  $x_{T-1}$  可表示成

$$\begin{aligned} x_T &= g_T \cdot \varepsilon_T \quad \text{where } g_T = 1 \\ x_{T-1} &= d_{T-1} \cdot x_T + e_{T-1} \cdot y_T + g_{T-1} \cdot \varepsilon_{T-1} \\ &= e_{T-1} y_T + (d_{T-1} g_T \varepsilon_T + g_{T-1} \varepsilon_{T-1}) \end{aligned} \tag{32}$$

根据  $x_{T-1}$  的结果, 可把  $x_{T-2}$  的结果写成

$$\begin{aligned} x_{T-2} &= d_{T-2} \cdot x_{T-1} + e_{T-2} \cdot y_{T-1} + g_{T-2} \cdot \varepsilon_{T-2} \\ &= (d_{T-2} e_{T-1} y_T + e_{T-2} y_{T-1}) + (d_{T-2} d_{T-1} g_T \varepsilon_T + d_{T-2} g_{T-1} \varepsilon_{T-1} + g_{T-2} \varepsilon_{T-2}) \end{aligned} \tag{33}$$

根据  $x_{T-2}$  的结果, 可进一步把  $x_{T-3}$  的结果写成

$$\begin{aligned} x_{T-3} &= d_{T-3} \cdot x_{T-2} + e_{T-3} \cdot y_{T-2} + g_{T-3} \cdot \varepsilon_{T-3} \\ &= (d_{T-3} d_{T-2} e_{T-1} y_T + d_{T-3} e_{T-2} y_{T-1} + e_{T-3} y_{T-2}) \\ &\quad + (d_{T-3} d_{T-2} d_{T-1} g_T \varepsilon_T + d_{T-3} d_{T-2} g_{T-1} \varepsilon_{T-1} + d_{T-3} g_{T-2} \varepsilon_{T-2} + g_{T-3} \varepsilon_{T-3}) \end{aligned} \tag{34}$$

类似地, 可递归把各个  $t$  对应的  $x_t$  写成类似的形式。可以看出, 每个  $x_t$  都可拆分成两部分, 一部分是过往预测  $x_0$ (即  $y_t$ ) 的加权和, 另一部分是过往噪声和新加噪声的加权和。由于  $d_t$ 、 $e_t$  和  $g_t$  均是已知的常量, 所以可以准确计算各个  $x_t$  的等效信号系数以及等效噪声系数。

计算结果显示, 各个  $x_t$  的等效信号系数几乎等于  $\sqrt{\bar{\alpha}_t}$ , 等效噪声系数约等于  $\sqrt{1 - \bar{\alpha}_t}$ , 并且仅有的微小误差还随着采样步数  $T$  的增大而减少, 具体可看图8, 分别列出了 18 步, 100 步, 500 步的结果。表格4以矩阵的形式给出了每个  $x_t$  关于  $y_t$  的完整系数, 每行对应一个  $x_t$ 。表格5以矩阵的形式给出了每个  $x_t$  关于  $\varepsilon_t$  的完整系数。可以看出, 噪声系数矩阵与信号系数矩阵有一点区别, 即对角元素的右侧多一个非零系数, 这说明每一步均有微量新噪声加入, 新噪声将使整体噪声的“模式”以缓慢的速度改变。

至此, 我们已成功证明, DDPM 祖先采样的过程可使用 Natural Inference 框架表示。

#### 4.4 Represent DDIM with Natural Inference Framework

DDIM 采样算法的迭代规则可表示成如下的形式:

$$\begin{aligned} y_t &= f_t(x_t) \\ x_{t-1} &= \sqrt{\bar{\alpha}_{t-1}} \cdot y_t + \sqrt{1 - \bar{\alpha}_{t-1}} \cdot \frac{x_t - \sqrt{\bar{\alpha}_t} y_t}{\sqrt{1 - \bar{\alpha}_t}} \cdot y_t \\ &= d_{t-1} \cdot x_t + e_{t-1} \cdot y_t \\ \text{where } d_{t-1} &= \frac{\sqrt{1 - \bar{\alpha}_{t-1}}}{\sqrt{1 - \bar{\alpha}_t}} \quad e_{t-1} = (\sqrt{\bar{\alpha}_{t-1}} - \frac{\sqrt{1 - \bar{\alpha}_{t-1}}}{\sqrt{1 - \bar{\alpha}_t} \sqrt{\bar{\alpha}_t}}) \end{aligned} \tag{35}$$

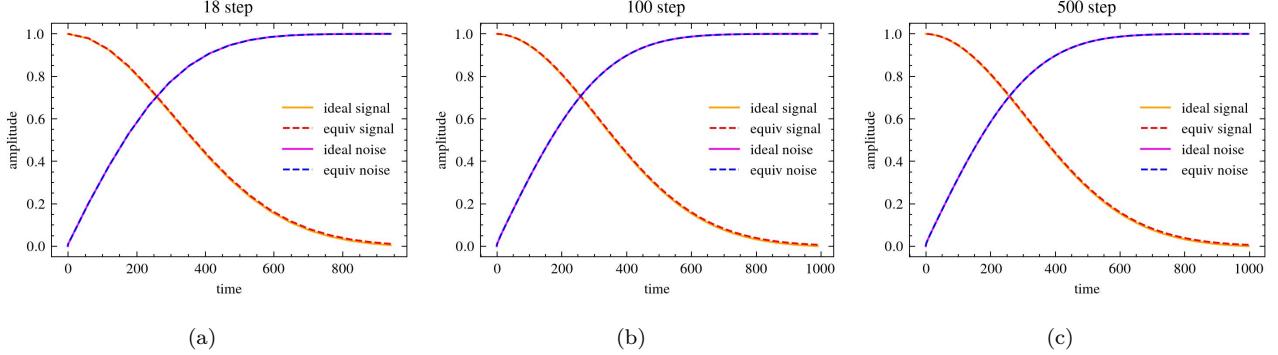


Figure 9: DDIM equivalent marginal coefficients and ideal marginal coefficients (a) 18 step (b) 100 step (c) 500 step

可以看出，DDIM 的更新公式与 DDPM 祖先采样的更新公式类似，只是少了  $g_{t-1} \cdot \varepsilon_{t-1}$  项，即每步不会增加新的噪声。仿照 DPPM 祖先采样的递推方法，也可把各个  $t$  对应的  $x_t$  写成类似的形式，如下：

$$\begin{aligned}
 x_T &= g_T \cdot \varepsilon_T \quad \text{where } g_T = 1 \\
 x_{T-1} &= e_{T-1} y_T + d_{T-1} g_T \varepsilon_T \\
 x_{T-2} &= (d_{T-2} e_{T-1} y_T + e_{T-2} y_{T-1}) + d_{T-2} d_{T-1} g_T \varepsilon_T \\
 x_{T-3} &= (d_{T-3} d_{T-2} e_{T-1} y_T + d_{T-3} e_{T-2} y_{T-1} + e_{T-3} y_{T-2}) \\
 &\quad + d_{T-3} d_{T-2} d_{T-1} g_T \varepsilon_T
 \end{aligned} \tag{36}$$

可以看出，DDIM 的形式与 DDPM 略微不同，由于 DDIM 每步不会新增噪声，所以只有一个噪声项。

计算结果显示，各个  $x_t$  的等效信号系数近似等于  $\sqrt{\alpha_t}$ ，等效噪声系数只有关于  $\varepsilon_T$  一项，其系数几乎等于  $\sqrt{1 - \bar{\alpha}_t}$ 。图9分别画出了 18 步，100 步，500 步的结果，可以看出，等效系数的误差很小，几乎看不出来。表格6给出了 18 步的完整信号系数矩阵。

因此，DDIM 的采样过程也可使用 Natural Inference 框架表示。

#### 4.5 Represent Flow Matching Euler Sampling with Natural Inference Framework

Flow Matching 的噪声混合方式如式(2)所示，当使用 Euler 离散积分采样时，其更新规则可表示成如下的形式：

$$\begin{aligned}
 y_i &= f_i(x_i) \\
 x_{i-1} &= x_i + (t_{i-1} - t_i)(-y_i + \epsilon) \\
 &= x_i + (t_{i-1} - t_i) \frac{x_i - y_i}{t_i} \\
 &= d_{i-1} \cdot x_i + e_{i-1} \cdot y_i \\
 \text{where } d_{i-1} &= \frac{t_{i-1}}{t_{t_i}} \quad e = (1 - \frac{t_{i-1}}{t_{t_i}})
 \end{aligned} \tag{37}$$

其中， $f$  是预测  $x_0$  的模型， $y_i$  是离散点  $t_i$  对应的模型  $f_i$  的输出值。可以看出，Flow Matching 的 Euler 算法的递推规则与 DDIM 类似，所以也可把各个  $x_i$  表示成类似的形式。

计算结果显示，对于每个离散点  $t_i$ ， $x_i$  的等效信号系数完全等于  $1 - t_i$ ，等效噪声系数只有  $\varepsilon_N$  一项，其系数完全等于  $t_i$ 。关于具体结果可看图10，分别列出了 18 步、200 步和 500 步的结果。表格7给出了 18 步的信号系数矩阵。

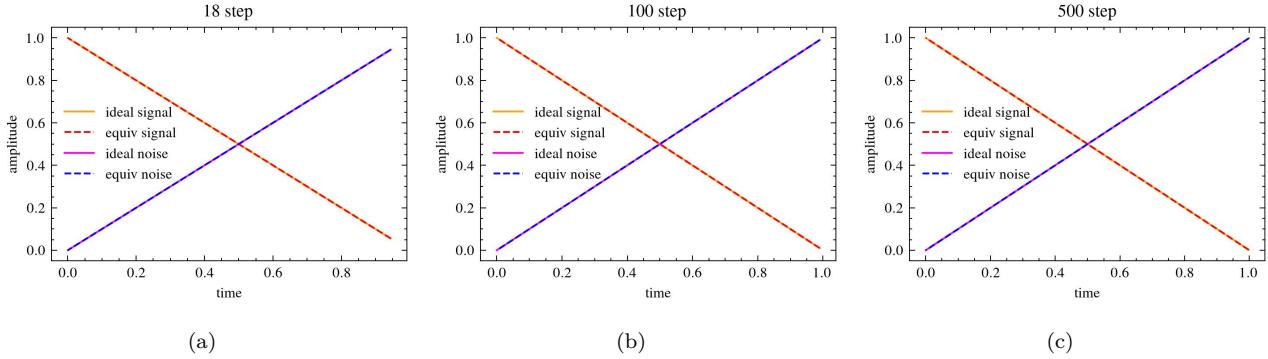


Figure 10: Flow matching euler sampler equivalent marginal coefficients and ideal marginal coefficients (a) 18 step (b) 100 step (c) 500 step

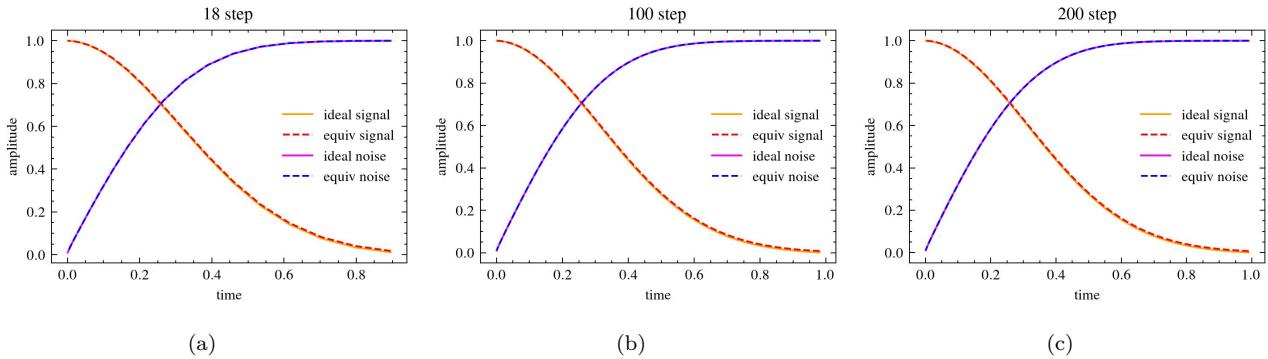


Figure 11: DEIS equivalent marginal coefficients and ideal marginal coefficients (a) 18 step (b) 100 step (c) 500 step

#### 4.6 Represent High Order Samplers with Natural Inference Framework

在前面几个小节，已经使用 Natural Inference 框架的形式表示了大多数一阶采样算法。对于二阶及更高的采样算法，由于  $x_i$  的更新规则比较复杂，使用上述的推导方法直接表示各个  $x_i$  的形式变得比较困难，于是，必须寻求其它解决方法。符号计算工具正好能解决这个痛点，它能自动分析复杂的数学表达式，只需在原有算法代码的基础上稍加改动，便能自动分析计算每个  $y_i$  项和  $\varepsilon_i$  项的系数。具体细节可参考本文章附录的代码。

经过实验计算后发现，DEIS、DPMSSolver 以及 DPMSSolver++ 均与 DDIM 存在同样的结论，每个  $x_i$  都可拆分成两部分，其信号等效系数约等于  $\sqrt{\alpha_i}$ ，噪声等效系数约等于  $\sqrt{1 - \alpha_i}$ 。

图11给出了 DEIS(tab3) 算法的结果，图12给出了三阶 DPMSSolver 的结果，图13给出了二阶 DPMSSolver++ 的结果，可以看出，这几个高阶采样算法也有着同样的性质，也都可以使用 Natural Inference 框架表示。表格8给出了三阶的 DEIS 算法的系数矩阵 (18 步)。表格9和表格10给出了二阶和三阶的 DPMSSolver 算法的系数矩阵 (18 步)。表格11和表格12给出了二阶和三阶的 DPMSSolver++ 算法的系数矩阵 (18 步)。

## 5 Application of the Natural Inference Framework

至此，我们已经使用了一种全新的视角来解释高维扩散模型，包括训练阶段的目标函数和测试阶段的推断算法。新视角具有多个优点：

- 新视角保持训练-测试一致性，训练阶段的目标是预测  $x_0$ ，测试阶段也是在预测  $x_0$ ，这有利于分析模型存在的问题以及设计问题的解决方案。
- 新视角把推断过程分成一系列预测  $x_0$  的操作，每个操作都有明确的输入图像信号和输出图像信号，这让推断过程变得可视化和可解释，对调试分析问题有明显的帮助。图18和图19提供了一个推断过程可视化的例子。

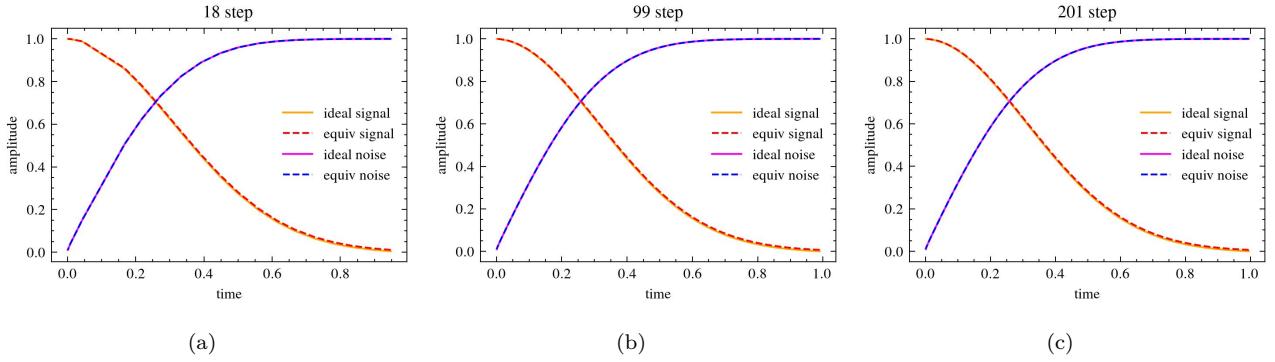


Figure 12: dpm solver3s equivalent marginal coefficients and ideal marginal coefficients (a) 18 step (b) 99 step (c) 201 step

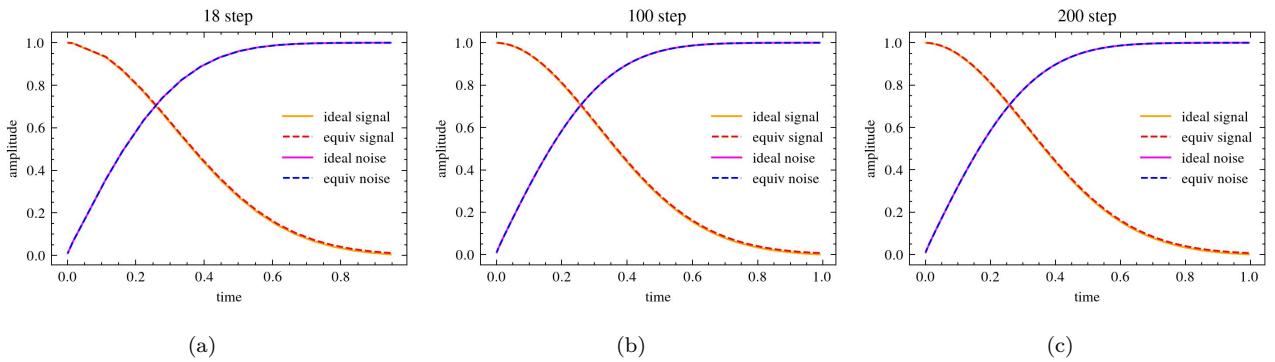


Figure 13: dpm solver++2s equivalent marginal coefficients and ideal marginal coefficients (a) 18 step (b) 99 step (c) 201 step

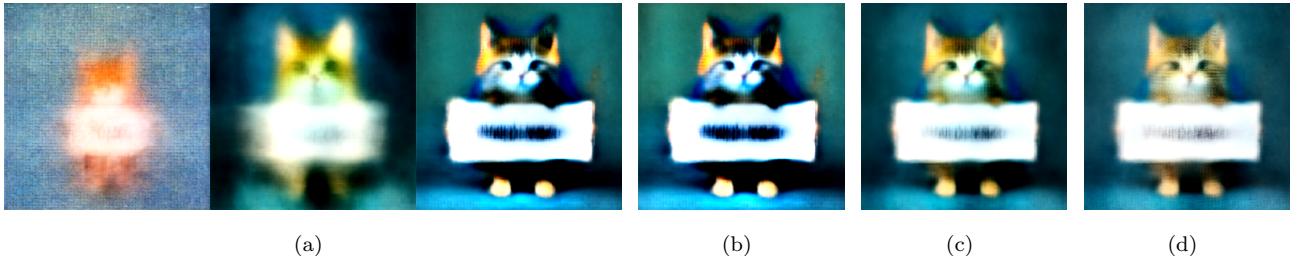


Figure 14: (a) the first three output( $x_0^0, x_0^1, x_0^2$ ) (b)  $0.00 \cdot x_0^0 - 0.17 \cdot x_0^1 + 1.17 \cdot x_0^2$  (c)  $0.00 \cdot x_0^0 + 0.49 \cdot x_0^1 + 0.51 \cdot x_0^2$  (d)  $0.32 \cdot x_0^0 + 0.33 \cdot x_0^1 + 0.35 \cdot x_0^2$

- 在新视角下，目前已有的采样算法只是 Natural Inference 框架的一种具体参数配置，并非最优。在下面的章节中将会看到，经过仔细的设计，可找到更优的参数配置。另外，新框架也提供了另外了一个自由度，通过控制 Self Guidance 的程度，可灵活控制生成图像的锐度。

### 5.1 Why can High Order Samplings Speedup Sampling ?

这一节，我们将以 Natural Inference 框架为基础，解释在低采样步数情况下，DEIS、DPMSSolver 等算法为什么比 DDPM 和 DDIM 表现更好。

表格4和表格6分别给出了在 Natural Inference 框架下的 DDPM 和 DDIM 系数矩阵，表格分别给出了 DEIS 和 DPMSSolver 的系数矩阵，仔细对比，可发现有一处明显的区别：DEIS 和 DPMSolver 的系数里存在负数，根据4.1的分析可知，其可视为 Fore Self Guidance 的操作，而 Fore Self Guidance 具有提升图像品质的功能；而 DDPM 和 DDIM 的系数均为正数，属于 Middle Self Guidance，没有提升效果的功能。因此，可以得出结论，高阶采样算法具有更好效果的关键在于 Self Guidance 的设计。

### 5.2 Why can Low order Sampling Adapt to Big CFG ?

我们知道，在较大 cfg 的情况下，大部分高阶采样算法会出现类似过暴的问题，包括 3 阶的 DEIS 和 DPMSSolver，而 DDIM 和 2 阶的 DPMSolver++ 却能缓解这种问题。那为什么它们可以，而其它算法也却不行呢？通过比较它们在 Natural Inference 框架下的系数矩阵，可以总结出问题的核心本质。表格6和表格11分别给出了 DDIM 和 2 阶的 DPMSolver++ 的系数矩阵，表格8和表格10分别给出了 3 阶的 DEIS 和 DPMSolver 的系数矩阵，比较可以看出，2 阶的 DPMSolver++ 和 DDIM 的系数均为正数，是 Middle Self Guidance 的复合操作，而 3 阶 DEIS 和 DPMSolver 存在负数，存在 Fore Self Guidance 的操作。所以，Mid Self Guidance(正系数) 是适应较大的 cfg 的关键。

图14给出了一个小实验，可证明：在较大的 cfg 情况下，Middle Self Guidance 的输出结果会明显优于 Fore Self Guidance。图14(a) 给出了 SD3 前三步的输出 (cfg=7)，图14(b)~(d) 分别给出了不同线性组合的结果，图14(b) 是带有负数系数的结果，图14(d) 是有两个正数系数的结果，图14(f) 是有三个正数的结果。可以看出，正数系数比较多时，图像效果最好。

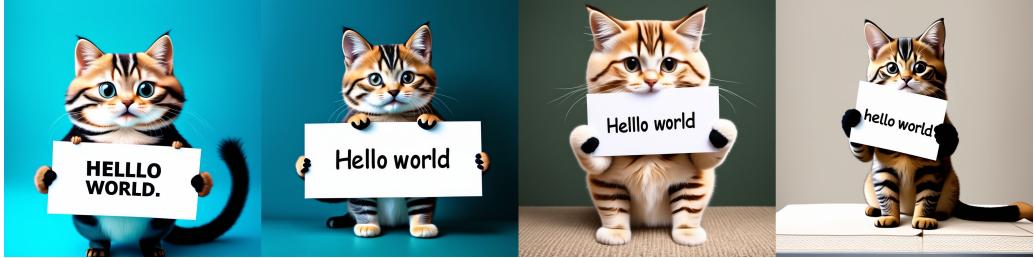
另外，我们知道，相对于 DDIM，2 阶 dpmsolver++ 不仅能适应较大 cfg，还具有相对较好的加速效果，这一点也可以从系数矩阵里找到解释，相对于 DDIM，2 阶 dpmsolver++ 的系数矩阵有更多的零项，更多的权重放在对角元素上，即当前步最近的输出上，所以会有更好的加速效果。

### 5.3 A Way to Control Image Sharpness

由4.1节可知，不同的 Self Guidance 操作会影响图像的锐度，因此，合理设计系数矩阵的权重分布，可控制生成图像的锐度。下面给出一个 SD3 例子，表格13给出了原始 euler 方法的系数矩阵，表格14给出了调整过的系数矩阵。可以看出，Euler 方法的 Self Guidance 操作基本都是 Mid Self Guidance 的复合操作，并且把不少的权重的放在较早的输出上；调整过后的系数矩阵，把较早输出的权重都置为 0，更多权重放在较近的输出上。



(a)



(b)

Figure 15: (a) Euler sampling result (d) Result with adjusted coefficient matrix

因此，调整后的系数矩阵能产生更锐利的图像，具体可看图15。注意，调整后系数矩阵虽然尽量减少较早输出的权重，但每行都至少保留三个非零项，这主要是为了适应较大的 cfg 的影响，特别是前面几步。

#### 5.4 Better Coefficient Matrix

由第4节的分析可知，目前主流的采样算法只是对应 Natural Inference 框架的一个系数矩阵，那是否有其它更高效的系数矩阵呢？答案是有的。下面将以 CIFAR10 数据集上的预训练模型为例子，证明在同样的步数下，存在其它更高效系数矩阵，能取得更高的 FID 指标。预训练模型采用的是 ScoreSDE[Son+20] 发布的 cifar10\_ddpmpp\_continuous(vp)。

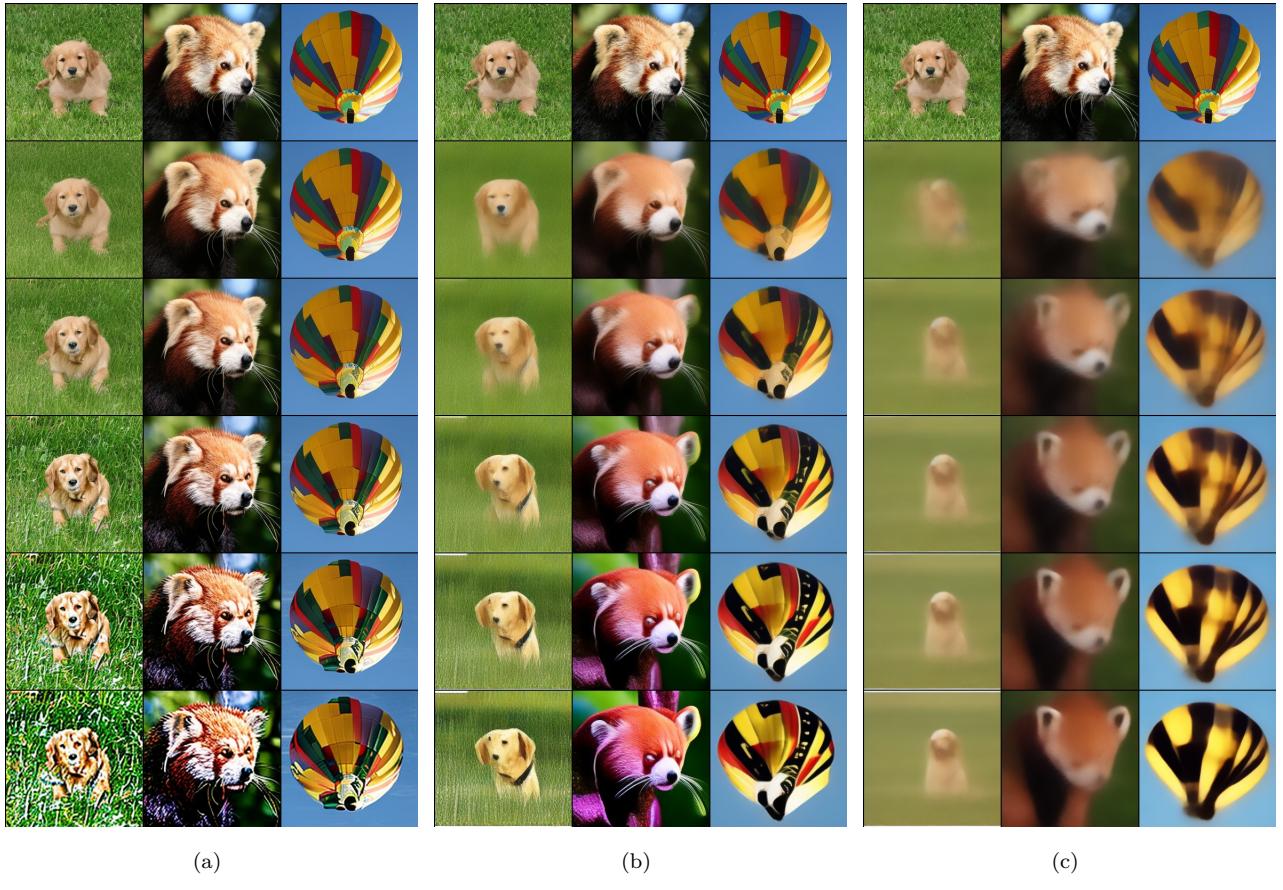
表格15给出了一个经过优化后的 5 步的系数矩阵，表格16给出了一个经过优化后的 10 步的系数矩阵，表格17给出了一个经过优化后的 15 步的系数矩阵。表格3给出了优化后的系数矩阵与 DEIS、DPMSSolver、DPMSSolver++ 的比较结果。可以看出，在同等步数下，优化后的系数矩阵比 DEIS、DPMSSolver 以及 DPMSSolver++ 有着更好的 FID 指标。每种算法均采用同样的时间离散策略，对于 DEIS、DPMSSolver 及 DPMSSolver++，遍历所有的超参数，并选择最低的 FID 作为结果。

系数矩阵的核心思想主要有两点：

- 权重分布倾向于当前时刻附近的输出，对较早的输出减少权重，也就是说，“减少拖尾的长度”。一般情况下，采样步数越多，需适当延长“拖尾”，从而避免“过增强”现象（见5.3节）；采样步数较少，可缩短“拖尾”。
- 在步数较少时，适当增加一些 Fore Self Guidance，即在对角元素的前面增加一些比较大的负的系数。

Table 3: Comparison of different algorithm

step\method	DEIS	DPMSSolver	DPMSSolver++	optimized matrix
5 step	15.60	13.16	11.50	<b>8.66</b>
10 step	3.94	5.06	4.64	<b>3.58</b>
15 step	3.55	4.15	3.92	<b>2.93</b>



(a)

(b)

(c)

Figure 16: over enhancing phenomenon on ddpm latent model trained on imageget-256 (a) on  $t=100$  (b)  $t=300$  (c)  $t=500$

## 5.5 Over Enhancing Phenomenon

这一小节介绍一种现象-“过增强”，这种现象影响着系数矩阵的设计。

在3.3中，我们知道拟合  $x_0$  实际上是在弥补被噪声淹没的高频分量，因此，会增加输入图像的细节，所以，可把模型操作视作一个图像质量增强算子。另外，由上面的分析也可知道，Self Guidance 操作也可增强图像的质量。当应用比较多“增强”操作时，“过增强”现象将容易出现。比如，频繁密集的模型增强(采样时间点比较密集， $t$  间距比较小)，或者 Self Guidance 操作的强度较大。

这种现象可通过下面的操作直观体现出来：给予一个正常的输入图像，固定模型时间点  $t$  和噪声，连续应用模型增强多次，“过增强现象”将逐渐显现。 $t$  是固定的，间距为 0，满足密集性的要求；模型输出的图像直接作为下一次模型的输入图像，可看作是没有“拖尾”且  $\lambda = 1$  的 Self Guidance，强度较大。具体效果看图16的过程，第一行是原始图像，第二行是第一次应用模型后的结果，可以看出图像质量有所下降，这主要是因为混合了噪声后，部分高频分量被淹没，有较小的信噪比，模型难以预测。从第四行开始，“过增强”现象逐渐显现，并越来越严重。另外，可以看出，不同的时间  $t$ ，“过增强”现象有不同的特点。 $t$  较小时，画面呈现突出“颗粒感”，有较多高频信息， $t$  较大时，图像变得简化，缺乏细节。

这种现象可能与模型输入的“训练-推断不匹配”有关系。在训练阶段，输入的图像是正常的图像，包含完整的频率成分，而在推理阶段，模型输入的图像并非是完整频率成分的，推断过程的早期，通常只有低频分量，随着  $t$  的增大，频率成分才逐渐增加，所以这两个数据域是存在一定差别的。

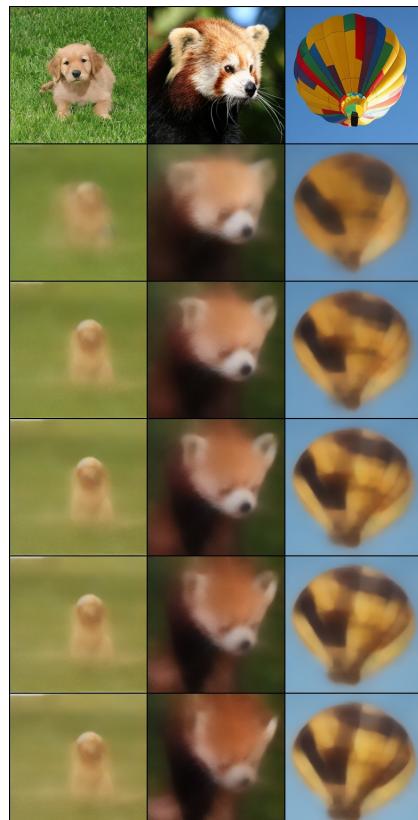
通过一个简单的实验，可验证此分析是否合理。对一个已经明显出现“过增强”的模型，在训练数据中混入部分类似图16第二和第三行的数据，进行微调，然后按照同样的方式生成“过增强”图像，结果如图17，可以看出，图像效果有明显变化： $t$  较小时，颗粒感明显减少； $t$  较大时，保留更多的细节。



(a)



(b)



(c)

Figure 17: improved over enhancing phenomenon after finetuning with model output images (a) on  $t=100$  (b)  $t=300$  (c)  $t=500$

## 5.6 Limitations and Directions for Improvement

虽然存在比现有采样方法更优的系数矩阵，但更优的系数矩阵并没有固定的形式，当模型、步数、时间离散策略改变时，系数矩阵可能也需要做相应性的调整。当步数比较多时，系数矩阵的可调整参数将比较多，会给人工调整带来一定挑战。一种自动化的解决方案是使用超参数优化方法自动搜索最优的配置 [YS20]；另一种是把 Self Guide 操作（线性加权和）替换成神经网络，单独优化或者与原有的模型参数进行联合优化。对于联合优化的方式，模型将呈现出自回归的形式，不管是训练阶段还是推断阶段。

## 6 Conclusion

本文主要尝试分析高维空间扩散模型的工作原理。从优化的目标函数出发，分析了高维数据稀疏性对目标函数的影响，以此为基础，提出了一种理解目标函数的新视角。同时，本文提出了一种新的推断框架，新框架不仅统一了现有主流的推断方法，还与目标函数的新视角相匹配。另外，新推断框架还能有效解释一些现象，同时也能指导设计更高效的推断方法。希望本文章能激发社区重新思考高维扩散模型的工作原理以及进一步改进训练和推理方式。

## References

- [And82] Brian DO Anderson. “Reverse-time diffusion equation models”. In: *Stochastic Processes and their Applications* 12.3 (1982), pp. 313–326.
- [Ban+23] Arpit Bansal et al. “Cold diffusion: Inverting arbitrary image transforms without noise”. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 41259–41282.
- [Bao+22a] Fan Bao et al. “Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models”. In: *arXiv preprint arXiv:2201.06503* (2022).
- [Bao+22b] Fan Bao et al. “Estimating the optimal covariance with imperfect mean in diffusion probabilistic models”. In: *arXiv preprint arXiv:2206.07309* (2022).
- [Bao+22c] Fan Bao et al. “Why are conditional generative models better than unconditional ones?” In: *arXiv preprint arXiv:2212.00362* (2022).
- [Bao+23] Fan Bao et al. “All are worth words: A vit backbone for diffusion models”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, pp. 22669–22679.
- [Die24] Sander Dieleman. *Diffusion is spectral autoregression*. 2024. URL: <https://sander.ai/2024/09/02/spectral-autoregression.html>.
- [DN21] Prafulla Dharwal and Alexander Nichol. “Diffusion models beat gans on image synthesis”. In: *Advances in neural information processing systems* 34 (2021), pp. 8780–8794.
- [Ess+24] Patrick Esser et al. “Scaling rectified flow transformers for high-resolution image synthesis”. In: *Forty-first international conference on machine learning*. 2024.
- [GW17] Rafael Gonzalez and Richard Woods. *Digital image processing, 4th Edition*. Pearson education india, 2017. Chap. 3.6, pp. 182–183.
- [HJA20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. “Denoising diffusion probabilistic models”. In: *Advances in neural information processing systems* 33 (2020), pp. 6840–6851.
- [HS22] Jonathan Ho and Tim Salimans. “Classifier-free diffusion guidance”. In: *arXiv preprint arXiv:2207.12598* (2022).
- [KW+13] Diederik P Kingma, Max Welling, et al. *Auto-encoding variational bayes*. 2013.

- [LGL22] Xingchao Liu, Chengyue Gong, and Qiang Liu. “Flow straight and fast: Learning to generate and transfer data with rectified flow”. In: *arXiv preprint arXiv:2209.03003* (2022).
- [Lip+22] Yaron Lipman et al. “Flow matching for generative modeling”. In: *arXiv preprint arXiv:2210.02747* (2022).
- [Lu+22a] Cheng Lu et al. “Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 5775–5787.
- [Lu+22b] Cheng Lu et al. “Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models”. In: *arXiv preprint arXiv:2211.01095* (2022).
- [ND21] Alexander Quinn Nichol and Prafulla Dhariwal. “Improved denoising diffusion probabilistic models”. In: *International conference on machine learning*. PMLR. 2021, pp. 8162–8171.
- [Pod+23] Dustin Podell et al. “Sdxl: Improving latent diffusion models for high-resolution image synthesis”. In: *arXiv preprint arXiv:2307.01952* (2023).
- [PX23] William Peebles and Saining Xie. “Scalable diffusion models with transformers”. In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2023, pp. 4195–4205.
- [Rad+21] Alec Radford et al. “Learning transferable visual models from natural language supervision”. In: *International conference on machine learning*. PmLR. 2021, pp. 8748–8763.
- [Rom+22] Robin Rombach et al. “High-resolution image synthesis with latent diffusion models”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 10684–10695.
- [Ros10] Sheldon Ross. “A first course in probability”. In: (2010).
- [SD+15] Jascha Sohl-Dickstein et al. “Deep unsupervised learning using nonequilibrium thermodynamics”. In: *International conference on machine learning*. pmlr. 2015, pp. 2256–2265.
- [SME20] Jiaming Song, Chenlin Meng, and Stefano Ermon. “Denoising diffusion implicit models”. In: *arXiv preprint arXiv:2010.02502* (2020).
- [Son+20] Yang Song et al. “Score-based generative modeling through stochastic differential equations”. In: *arXiv preprint arXiv:2011.13456* (2020).
- [Vin11] Pascal Vincent. “A connection between score matching and denoising autoencoders”. In: *Neural computation* 23.7 (2011), pp. 1661–1674.
- [YS20] Li Yang and Abdallah Shami. “On hyperparameter optimization of machine learning algorithms: Theory and practice”. In: *Neurocomputing* 415 (2020), pp. 295–316.
- [ZC22] Qinsheng Zhang and Yongxin Chen. “Fast sampling of diffusion models with exponential integrator”. In: *arXiv preprint arXiv:2204.13902* (2022).
- [Zhe23] zhenxin Zheng. *The Art of DPM*. 2023. URL: [https://github.com/blairstar/The\\_Art\\_of\\_DPM](https://github.com/blairstar/The_Art_of_DPM).
- [Zhe24] zhenxin Zheng. *Understanding Diffusion Probability Model Interactively*. 2024. URL: <https://huggingface.co/spaces/blairzheng/DPMInteractive>.

## A Some Proofs

### A.1 Predicting Posterior Mean is Equivalent to Predicting $X_0$

下面证明下述两个目标函数是等价的：

$$\min_{\theta} \int p(x_t) \left\| f_{\theta}(x_t) - \int p(x_0|x_t) x_0 dx_0 \right\|^2 dx_t \iff \min_{\theta} \iint p(x_0, x_t) \|f_{\theta}(x_t) - x_0\|^2 dx_0 dx_t \quad (38)$$

Proof:

对于  $\|f_\theta(x_t) - \int p(x_0|x_t)x_0dx_0\|^2$ , 存在如下的关系:

$$\left\|f_\theta(x_t) - \int p(x_0|x_t)x_0dx_0\right\|^2 = f_\theta^2(x_t) - 2f_\theta(x_t) \int p(x_0|x_t)x_0dx_0 + \left\|\int p(x_0|x_t)x_0dx_0\right\|^2 \quad (39)$$

$$= \int p(x_0|x_t)f_\theta^2(x_t)dx_0 - 2f_\theta(x_t) \int p(x_0|x_t)x_0dx_0 + C_1 \quad (40)$$

$$= \int p(x_0|x_t) (f_\theta^2(x_t) - 2f_\theta(x_t)x_0 + x_0^2) dx_0 - \int p(x_0|x_t)x_0^2dx_0 + C_1 \quad (41)$$

$$= \int p(x_0|x_t) \|f_\theta^2(x_t) - x_0\|^2 dx_0 - C_2 + C_1 \quad (42)$$

其中,  $C_1$  和  $C_2$  是不依赖  $\theta$  的固定值。

把上述关系代入预测均值的目标函数, 可得

$$\int p(x_t) \left\|f_\theta(x_t) - \int p(x_0|x_t)x_0dx_0\right\|^2 dx_t = \int p(x_t) \left( \int p(x_0|x_t) \|f_\theta^2(x_t) - x_0\|^2 dx_0 - C_2 + C_1 \right) dx_t \quad (43)$$

$$= \iint p(x_0, x_t) \|f_\theta(x_t) - x_0\|^2 dx_0 dx_t + \int p(x_t) (C_1 - C_2) dx_t \quad (44)$$

$$= \iint p(x_0, x_t) \|f_\theta(x_t) - x_0\|^2 dx_0 dx_t + C_3 \quad (45)$$

也就是说, 两个目标函数只差一个与优化参数无关的常量。于是, 两个目标函数是等价的。

## A.2 Conditional Score

下面证明如下的关系:

$$\frac{\partial \log p(x_t)}{\partial x_t} = \int p(x_0|x_t) \frac{\partial \log p(x_t|x_0)}{\partial x_t} dx_0 \quad (46)$$

Proof:

$$\frac{\partial \log p(x_t)}{\partial x_t} = \frac{1}{p(x_t)} \frac{\partial p(x_t)}{\partial x_t} \quad (47)$$

$$= \frac{1}{p(x_t)} \frac{\partial (\int p(x_0)p(x_t|x_0)dx_0)}{\partial x_t} \quad (48)$$

$$= \int \frac{p(x_0)}{p(x_t)} \frac{\partial p(x_t|x_0)}{\partial x_t} dx_0 \quad (49)$$

$$= \int \frac{p(x_0, x_t)/p(x_t)}{p(x_0, x_t)/p(x_0)} \frac{\partial p(x_t|x_0)}{\partial x_t} dx_0 \quad (50)$$

$$= \int \frac{p(x_0|x_t)}{p(x_t|x_0)} \frac{\partial p(x_t|x_0)}{\partial x_t} dx_0 \quad (51)$$

$$= \int p(x_0|x_t) \frac{\partial \log p(x_t|x_0)}{\partial x_t} dx_0 \quad (52)$$

## A.3 Form of the Posterior Probability

下面的推导参考了 [Zhe23] 和 [Zhe24] 的内容。

假设  $x_t$  有如下的形式:

$$x_t = c_0 \cdot x_0 + c_1 \cdot \epsilon \quad \text{where } c_0 \text{ and } c_1 \text{ 是常数} \quad (53)$$

则有

$$p(x_t|x_0) \sim \mathcal{N}(x_t; c_0 x_0, c_1^2) \quad (54)$$

根据 Bayes 公式，有

$$p(x_0|x_t) = \frac{p(x_t|x_0)p(x_0)}{p(x_t)} \quad (55)$$

$$= \frac{p(x_t|x_0)p(x_0)}{\int p(x_t|x_0)p(x_0)dx_0} \quad (56)$$

$$= \text{Normalize}(p(x_t|x_0)p(x_0)) \quad (57)$$

其中，Normalize 表示归一化算子，归一化除数为  $\int p(x_t|x_0)p(x_0)dx_0$ 。

把式(54)代入，可得

$$p(x_0|x_t) = \text{Normalize} \left( \frac{1}{\sqrt{2\pi c_1^2}} \exp \frac{-(x_t - c_0 x_0)^2}{2c_1^2} p(x_0) \right) \quad (58)$$

$$= \text{Normalize} \left( \frac{1}{\sqrt{2\pi c_1^2}} \exp \frac{-(x_0 - \frac{x_t}{c_0})^2}{2\frac{c_1^2}{c_0^2}} p(x_0) \right) \quad (59)$$

$$= \text{Normalize} \left( \exp \frac{-(x_0 - \mu)^2}{2\sigma^2} p(x_0) \right) \quad (60)$$

$$\text{where } \mu = \frac{x_t}{c_0} \quad \sigma = \frac{c_1}{c_0} \quad (61)$$

上式推导中，由于归一化算子的存在，可把  $\frac{1}{\sqrt{2\pi c_1^2}}$  忽略。

## B Coefficient Matrixs

### B.1 DDPM Coefficient Matrix

Table 4: DDPM's signal coefficient matrix on natural inference framework

time	940	881	823	764	705	646	588	529	470	411	353	294	235	176	118	059	000	-01	sum
940	0.008	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.008	
881	0.005	0.013	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.017	
823	0.003	0.008	0.02	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.031	
764	0.002	0.005	0.013	0.032	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.051	
705	0.001	0.003	0.008	0.02	0.047	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.079	
646	0.001	0.002	0.005	0.013	0.031	0.067	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.119	
588	0.0	0.001	0.004	0.009	0.021	0.046	0.09	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.172	
529	0.0	0.001	0.003	0.006	0.015	0.032	0.062	0.12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.24	
470	0.0	0.001	0.002	0.005	0.01	0.022	0.044	0.085	0.154	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.323	
411	0.0	0.0	0.001	0.003	0.007	0.016	0.031	0.06	0.109	0.192	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.42	
353	0.0	0.0	0.001	0.002	0.005	0.011	0.022	0.042	0.076	0.135	0.232	0.0	0.0	0.0	0.0	0.0	0.0	0.526	
294	0.0	0.0	0.001	0.002	0.003	0.007	0.015	0.028	0.051	0.091	0.156	0.284	0.0	0.0	0.0	0.0	0.0	0.639	
235	0.0	0.0	0.0	0.001	0.002	0.005	0.009	0.018	0.033	0.057	0.099	0.18	0.345	0.0	0.0	0.0	0.0	0.749	
176	0.0	0.0	0.0	0.001	0.001	0.003	0.005	0.01	0.018	0.032	0.056	0.101	0.195	0.426	0.0	0.0	0.0	0.849	
118	0.0	0.0	0.0	0.0	0.001	0.001	0.002	0.005	0.008	0.015	0.026	0.047	0.09	0.196	0.536	0.0	0.0	0.927	
059	0.0	0.0	0.0	0.0	0.0	0.001	0.001	0.002	0.004	0.007	0.013	0.024	0.053	0.145	0.728	0.0	0.0	0.98	
000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.002	0.998	0.0	1.0	
-01	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	

Table 5: DDPM's noise coefficient matrix on natural inference framework

time	999	940	881	823	764	705	646	588	529	470	411	353	294	235	176	118	059	000	-01	norm
940	0.561	0.828	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	
881	0.326	0.481	0.814	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
823	0.197	0.292	0.494	0.795	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.999
764	0.123	0.181	0.307	0.494	0.782	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.999
705	0.079	0.117	0.197	0.318	0.502	0.763	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.997
646	0.052	0.077	0.131	0.211	0.333	0.506	0.741	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.993
588	0.036	0.053	0.09	0.144	0.228	0.347	0.508	0.712	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.985
529	0.025	0.037	0.062	0.1	0.159	0.241	0.353	0.496	0.687	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.971
470	0.018	0.026	0.044	0.071	0.112	0.17	0.249	0.349	0.485	0.653	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.946
411	0.012	0.018	0.031	0.05	0.079	0.12	0.176	0.247	0.342	0.462	0.613	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.907
353	0.009	0.013	0.022	0.035	0.056	0.084	0.123	0.173	0.24	0.324	0.43	0.564	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.85
294	0.006	0.009	0.015	0.024	0.037	0.057	0.083	0.117	0.162	0.218	0.29	0.38	0.513	0.0	0.0	0.0	0.0	0.0	0.0	0.769
235	0.004	0.005	0.009	0.015	0.024	0.036	0.053	0.074	0.102	0.138	0.183	0.24	0.324	0.449	0.0	0.0	0.0	0.0	0.0	0.662
176	0.002	0.003	0.005	0.008	0.013	0.02	0.03	0.042	0.058	0.078	0.103	0.135	0.183	0.253	0.375	0.0	0.0	0.0	0.0	0.529
118	0.001	0.001	0.002	0.004	0.006	0.009	0.014	0.019	0.027	0.036	0.048	0.062	0.084	0.117	0.173	0.285	0.0	0.0	0.0	0.375
059	0.0	0.0	0.001	0.001	0.002	0.003	0.004	0.005	0.007	0.01	0.013	0.017	0.023	0.032	0.047	0.077	0.173	0.0	0.0	0.201
000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.01	0.0	0.01	
-01	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

## B.2 DDIM Coefficient Matrix

Table 6: DDIM's signal coefficient matrix on the natural inference framework

time	940	881	823	764	705	646	588	529	470	411	353	294	235	176	118	059	000	-01	sum
940	0.005	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.005	
881	0.005	0.008	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.013
823	0.005	0.008	0.013	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.026
764	0.005	0.008	0.013	0.019	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.045
705	0.005	0.008	0.013	0.019	0.028	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.074
646	0.005	0.008	0.013	0.019	0.028	0.04	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.113
588	0.005	0.008	0.012	0.019	0.028	0.04	0.053	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.166
529	0.005	0.008	0.012	0.019	0.028	0.039	0.052	0.07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.234
470	0.005	0.008	0.012	0.018	0.027	0.038	0.051	0.069	0.089	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.317
411	0.005	0.007	0.011	0.018	0.026	0.037	0.049	0.066	0.086	0.111	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.415
353	0.004	0.007	0.011	0.017	0.024	0.034	0.046	0.062	0.08	0.104	0.132	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.521
294	0.004	0.006	0.01	0.015	0.022	0.031	0.041	0.056	0.073	0.094	0.12	0.163	0.0	0.0	0.0	0.0	0.0	0.0	0.634
235	0.003	0.005	0.008	0.013	0.019	0.027	0.036	0.048	0.063	0.081	0.103	0.14	0.199	0.0	0.0	0.0	0.0	0.0	0.745
176	0.003	0.004	0.007	0.01	0.015	0.021	0.029	0.038	0.05	0.065	0.082	0.112	0.159	0.25	0.0	0.0	0.0	0.0	0.845
118	0.002	0.003	0.005	0.007	0.011	0.015	0.02	0.027	0.035	0.046	0.058	0.08	0.113	0.177	0.325	0.0	0.0	0.0	0.924
059	0.001	0.002	0.003	0.004	0.006	0.008	0.011	0.015	0.019	0.025	0.031	0.043	0.06	0.095	0.174	0.483	0.0	0.0	0.978
000	0.0	0.0	0.0	0.0	0.0	0.0	0.001	0.001	0.001	0.001	0.002	0.002	0.003	0.005	0.009	0.024	0.951	0.0	1.0
-01	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	

## B.3 Flow Matching Coefficient Matrix

Table 7: Flow matching euler sampler's signal coefficient matrix on natural inference framework

time	0.944	0.889	0.833	0.778	0.722	0.667	0.611	0.556	0.500	0.444	0.389	0.333	0.278	0.222	0.167	0.111	0.056	0.000	sum
0.944	0.056	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.056	
0.889	0.052	0.059	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.111	
0.833	0.049	0.055	0.062	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.167	
0.778	0.046	0.051	0.058	0.067	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.222	
0.722	0.042	0.048	0.054	0.062	0.071	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.278	
0.667	0.039	0.044	0.05	0.057	0.066	0.077	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.333	
0.611	0.036	0.04	0.046	0.052	0.06	0.071	0.083	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.389	
0.556	0.033	0.037	0.042	0.048	0.055	0.064	0.076	0.091	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.444	
0.500	0.029	0.033	0.038	0.043	0.049	0.058	0.068	0.082	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	
0.444	0.026	0.029	0.033	0.038	0.044	0.051	0.061	0.073	0.089	0.111	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.556	
0.389	0.023	0.026	0.029	0.033	0.038	0.045	0.053	0.064	0.078	0.097	0.125	0.0	0.0	0.0	0.0	0.0	0.0	0.611	
0.333	0.02	0.022	0.025	0.029	0.033	0.038	0.045	0.055	0.067	0.083	0.107	0.143	0.0	0.0	0.0	0.0	0.0	0.667	
0.278	0.016	0.018	0.021	0.024	0.027	0.032	0.038	0.045	0.056	0.069	0.089	0.119	0.167	0.0	0.0	0.0	0.0	0.722	
0.222	0.013	0.015	0.017	0.019	0.022	0.026	0.03	0.036	0.044	0.056	0.071	0.095	0.133	0.2	0.0	0.0	0.0	0.778	
0.167	0.01	0.011	0.012	0.014	0.016	0.019	0.023	0.027	0.033	0.042	0.054	0.071	0.1	0.15	0.25	0.0	0.0	0.833	
0.111	0.007	0.007	0.008	0.01	0.011	0.013	0.015	0.018	0.022	0.028	0.036	0.048	0.067	0.1	0.167	0.333	0.0	0.0	0.889
0.056	0.003	0.004	0.004	0.005	0.005	0.006	0.008	0.009	0.011	0.014	0.018	0.024	0.033	0.05	0.083	0.167	0.5	0.0	0.944
0.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	

## B.4 DEIS Coefficient Matrix

Table 8: DEIS sampler's signal coefficient matrix on natural inference framework

time	0.895	0.796	0.703	0.616	0.534	0.459	0.389	0.324	0.266	0.213	0.167	0.126	0.090	0.061	0.037	0.019	0.007	0.001	sum
0.895	0.011	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.011
0.796	0.002	0.033	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.034
0.703	0.014	-0.01	0.072	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.076
0.616	-0.005	0.058	-0.043	0.13	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.14
0.534	0.014	-0.013	0.09	-0.046	0.183	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.229
0.459	-0.004	0.054	-0.037	0.135	-0.046	0.235	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.337
0.389	0.011	-0.005	0.069	-0.02	0.165	-0.046	0.283	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.457
0.324	-0.001	0.038	-0.015	0.093	-0.004	0.19	-0.047	0.324	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.577
0.266	0.007	0.004	0.041	0.004	0.105	0.009	0.209	-0.053	0.363	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.689
0.213	0.001	0.023	-0.001	0.055	0.017	0.113	0.016	0.223	-0.063	0.401	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.785
0.167	0.004	0.006	0.022	0.012	0.06	0.025	0.116	0.015	0.234	-0.076	0.441	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.86
0.126	0.001	0.013	0.003	0.03	0.018	0.062	0.026	0.117	0.009	0.245	-0.094	0.487	0.0	0.0	0.0	0.0	0.0	0.0	0.916
0.090	0.002	0.005	0.011	0.01	0.032	0.02	0.06	0.021	0.115	-0.003	0.257	-0.115	0.541	0.0	0.0	0.0	0.0	0.0	0.954
0.061	0.001	0.006	0.002	0.015	0.011	0.03	0.016	0.056	0.012	0.114	-0.02	0.271	-0.141	0.606	0.0	0.0	0.0	0.0	0.977
0.037	0.001	0.002	0.005	0.004	0.014	0.009	0.027	0.01	0.051	-0.0	0.112	-0.042	0.284	-0.173	0.687	0.0	0.0	0.0	0.99
0.019	0.0	0.002	0.001	0.006	0.004	0.012	0.005	0.022	0.002	0.045	-0.014	0.11	-0.066	0.292	-0.208	0.785	0.0	0.0	0.997
0.007	0.0	0.0	0.002	0.001	0.004	0.002	0.008	0.001	0.017	-0.005	0.039	-0.027	0.103	-0.088	0.285	-0.244	0.902	0.0	0.999
0.001	-0.0	0.0	-0.0	0.001	-0.0	0.002	-0.001	0.005	-0.003	0.012	-0.012	0.033	-0.039	0.09	-0.111	0.262	-0.319	1.078	1.0

## B.5 DPMsolver Coefficient Matrix

Table 9: DPMSSolver2S's signal coefficient matrix on natural inference framework

time	0.946	0.889	0.835	0.778	0.724	0.667	0.614	0.556	0.502	0.445	0.390	0.334	0.277	0.223	0.161	0.112	0.016	0.001	sum
0.946	0.005	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.005	
0.889	-0.008	0.021	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.012	
0.835	-0.008	0.021	0.011	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.023	
0.778	-0.008	0.021	-0.017	0.045	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.041	
0.724	-0.008	0.021	-0.017	0.045	0.024	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.064	
0.667	-0.008	0.02	-0.017	0.045	-0.029	0.088	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.099	
0.614	-0.008	0.02	-0.017	0.045	-0.029	0.087	0.044	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.143	
0.556	-0.008	0.02	-0.017	0.045	-0.029	0.086	-0.044	0.149	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.203	
0.502	-0.008	0.02	-0.016	0.044	-0.028	0.085	-0.043	0.146	0.073	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.272	
0.445	-0.008	0.019	-0.016	0.042	-0.027	0.082	-0.042	0.142	-0.059	0.225	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.359	
0.390	-0.007	0.018	-0.015	0.04	-0.026	0.078	-0.04	0.135	-0.056	0.215	0.112	0.0	0.0	0.0	0.0	0.0	0.0	0.454	
0.334	-0.007	0.017	-0.014	0.038	-0.024	0.073	-0.037	0.126	-0.052	0.2	-0.077	0.318	0.0	0.0	0.0	0.0	0.0	0.559	
0.277	-0.006	0.015	-0.012	0.034	-0.022	0.065	-0.033	0.112	-0.047	0.179	-0.069	0.285	0.168	0.0	0.0	0.0	0.0	0.669	
0.223	-0.005	0.013	-0.011	0.029	-0.019	0.056	-0.028	0.097	-0.04	0.154	-0.059	0.245	-0.112	0.45	0.0	0.0	0.0	0.768	
0.161	-0.004	0.01	-0.008	0.022	-0.014	0.043	-0.022	0.074	-0.031	0.118	-0.046	0.188	-0.086	0.346	0.278	0.0	0.0	0.869	
0.112	-0.003	0.007	-0.006	0.016	-0.01	0.031	-0.016	0.054	-0.023	0.086	-0.033	0.137	-0.063	0.252	-0.235	0.735	0.0	0.932	
0.016	-0.001	0.001	-0.001	0.003	-0.002	0.006	-0.003	0.01	-0.004	0.015	-0.006	0.024	-0.011	0.045	-0.042	0.13	0.833	0.0	0.998
0.001	-0.0	0.0	-0.0	0.0	-0.0	0.001	-0.0	0.002	-0.001	0.003	-0.001	0.004	-0.002	0.007	-0.007	0.022	-4.895	5.867	1.0

Table 10: DPMSSolver3S's signal coefficient matrix on natural inference framework

time	0.948	0.892	0.834	0.782	0.727	0.667	0.615	0.560	0.500	0.447	0.391	0.334	0.273	0.217	0.167	0.044	0.009	0.001	sum
0.948	0.004	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.004	
0.892	-0.004	0.016	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.012	
0.834	0.019	-0.033	0.037	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.024	
0.782	0.019	-0.033	0.037	0.016	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.039	
0.727	0.019	-0.033	0.037	-0.012	0.052	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.063	
0.667	0.019	-0.033	0.037	0.049	-0.078	0.104	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.099	
0.615	0.019	-0.033	0.037	0.049	-0.077	0.104	0.042	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.141	
0.560	0.019	-0.032	0.036	0.048	-0.076	0.103	-0.024	0.125	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.198	
0.500	0.019	-0.032	0.036	0.047	-0.075	0.101	0.093	-0.134	0.219	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.274	
0.447	0.018	-0.031	0.035	0.046	-0.073	0.098	0.09	-0.13	0.213	0.089	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.356	
0.391	0.017	-0.029	0.033	0.044	-0.069	0.093	0.086	-0.124	0.203	-0.04	0.238	0.0	0.0	0.0	0.0	0.0	0.0	0.452	
0.334	0.016	-0.027	0.031	0.041	-0.064	0.087	0.08	-0.115	0.188	0.147	-0.191	0.368	0.0	0.0	0.0	0.0	0.0	0.559	
0.273	0.014	-0.024	0.027	0.036	-0.057	0.077	0.071	-0.102	0.167	0.131	-0.17	0.327	0.178	0.0	0.0	0.0	0.0	0.675	
0.217	0.012	-0.02	0.023	0.031	-0.049	0.065	0.06	-0.087	0.142	0.111	-0.144	0.277	-0.078	0.435	0.0	0.0	0.0	0.778	
0.167	0.01	-0.017	0.019	0.025	-0.039	0.053	0.049	-0.07	0.115	0.09	-0.117	0.225	0.248	-0.336	0.605	0.0	0.0	0.859	
0.044	0.003	-0.005	0.006	0.007	-0.012	0.016	0.015	-0.021	0.035	0.027	-0.035	0.068	0.074	-0.101	0.181	0.73	0.0	0.987	
0.009	0.001	-0.001	0.001	0.002	-0.003	0.004	0.004	-0.006	0.009	0.007	-0.009	0.018	0.02	-0.027	0.048	-1.201	2.132	0.0	0.999
0.001	0.0	-0.0	0.0	0.001	-0.001	0.001	0.0	-0.001	0.002	-0.002	0.005	0.005	-0.007	0.013	6.607	-10.588	4.963	1.0	

## B.6 DPMSSolver++ Coefficient Matrix

Table 11: DPMSSolverpp2S's signal coefficient matrix on natural inference framework

time	0.946	0.889	0.835	0.778	0.724	0.667	0.614	0.556	0.502	0.445	0.390	0.334	0.277	0.223	0.161	0.112	0.016	0.001	sum
0.946	0.005	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.005	
0.889	0.0	0.012	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.012	
0.835	0.0	0.012	0.011	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.023	
0.778	0.0	0.012	0.0	0.029	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.041	
0.724	0.0	0.012	0.0	0.029	0.024	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.064	
0.667	0.0	0.012	0.0	0.028	0.0	0.059	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.099	
0.614	0.0	0.012	0.0	0.028	0.0	0.058	0.044	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.143	
0.556	0.0	0.012	0.0	0.028	0.0	0.058	0.0	0.105	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.203	
0.502	0.0	0.012	0.0	0.028	0.0	0.057	0.0	0.103	0.073	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.272	
0.445	0.0	0.011	0.0	0.027	0.0	0.055	0.0	0.1	0.0	0.166	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.359	
0.390	0.0	0.011	0.0	0.025	0.0	0.052	0.0	0.095	0.0	0.159	0.112	0.0	0.0	0.0	0.0	0.0	0.0	0.454	
0.334	0.0	0.01	0.0	0.024	0.0	0.049	0.0	0.089	0.0	0.147	0.0	0.241	0.0	0.0	0.0	0.0	0.0	0.559	
0.277	0.0	0.009	0.0	0.021	0.0	0.044	0.0	0.079	0.0	0.132	0.0	0.216	0.168	0.0	0.0	0.0	0.0	0.669	
0.223	0.0	0.008	0.0	0.018	0.0	0.037	0.0	0.068	0.0	0.113	0.0	0.185	0.0	0.338	0.0	0.0	0.0	0.768	
0.161	0.0	0.006	0.0	0.014	0.0	0.029	0.0	0.052	0.0	0.087	0.0	0.143	0.0	0.26	0.278	0.0	0.0	0.869	
0.112	0.0	0.004	0.0	0.001	0.0	0.021	0.0	0.038	0.0	0.064	0.0	0.104	0.0	0.189	0.0	0.501	0.0	0.0	

Table 12: DPMSolverpp3S's signal coefficient matrix on natural inference framework

time	0.948	0.892	0.834	0.782	0.727	0.667	0.615	0.560	0.500	0.447	0.391	0.334	0.273	0.217	0.167	0.044	0.009	0.001	sum
<b>0.948</b>	0.004	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.004
<b>0.892</b>	0.025	-0.014	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.012
<b>0.834</b>	0.046	0.0	-0.022	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.024
<b>0.782</b>	0.046	0.0	-0.022	0.016	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.039
<b>0.727</b>	0.046	0.0	-0.022	0.085	-0.045	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.063
<b>0.667</b>	0.046	0.0	-0.022	0.144	0.0	-0.068	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.099
<b>0.615</b>	0.045	0.0	-0.022	0.143	0.0	-0.068	0.042	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.141
<b>0.560</b>	0.045	0.0	-0.022	0.142	0.0	-0.067	0.211	-0.111	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.198
<b>0.500</b>	0.044	0.0	-0.021	0.139	0.0	-0.066	0.334	0.0	-0.156	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.274
<b>0.447</b>	0.043	0.0	-0.021	0.135	0.0	-0.064	0.325	0.0	-0.151	0.089	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.356
<b>0.391</b>	0.041	0.0	-0.02	0.129	0.0	-0.061	0.31	0.0	-0.144	0.415	-0.217	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.452
<b>0.334</b>	0.038	0.0	-0.018	0.119	0.0	-0.057	0.288	0.0	-0.134	0.6	0.0	-0.277	0.0	0.0	0.0	0.0	0.0	0.0	0.559
<b>0.273</b>	0.034	0.0	-0.016	0.106	0.0	-0.05	0.255	0.0	-0.119	0.533	0.0	-0.246	0.178	0.0	0.0	0.0	0.0	0.0	0.675
<b>0.217</b>	0.029	0.0	-0.014	0.09	0.0	-0.043	0.217	0.0	-0.101	0.452	0.0	-0.209	0.749	-0.393	0.0	0.0	0.0	0.0	0.778
<b>0.167</b>	0.023	0.0	-0.011	0.073	0.0	-0.035	0.176	0.0	-0.082	0.368	0.0	-0.17	0.962	0.0	-0.445	0.0	0.0	0.0	0.859
<b>0.044</b>	0.007	0.0	-0.003	0.022	0.0	-0.01	0.053	0.0	-0.025	0.11	0.0	-0.051	0.288	0.0	-0.133	0.73	0.0	0.0	0.987
<b>0.009</b>	0.002	0.0	-0.001	0.006	0.0	-0.003	0.014	0.0	-0.007	0.029	0.0	-0.013	0.076	0.0	-0.035	2.235	-1.304	0.0	0.999
<b>0.001</b>	0.0	0.0	-0.0	0.002	0.0	-0.001	0.004	0.0	-0.002	0.008	0.0	-0.004	0.02	0.0	-0.009	2.116	0.0	-1.134	1.0

## C SD3's Coefficient Matrix and Inference Process Visualization

### C.1 Coefficient Matrix

Table 13: SD3's flow euler signal coefficient matrix on natural inference framework

注意，为了方便阅读，表13的系数是原始系数乘以 100 后的结果。使用时，需归一化至每步相应的 Marginal Coefficient，例如，第一行需归一化至 0.0126，第二行需归一化至 0.0259。表14的使用也是如此。

Table 14: SD3's signal coefficient matrix with more sharpness

time	0.99	0.97	0.96	0.95	0.93	0.91	0.90	0.88	0.86	0.84	0.81	0.79	0.76	0.74	0.71	0.68	0.64	0.60	0.56	0.52	0.46	0.41	0.35	0.28	0.20	0.11	0.01	0.00
<b>0.99</b>	1.26	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
<b>0.97</b>	1.26	1.33	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
<b>0.96</b>	1.26	1.33	1.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
<b>0.95</b>	0.0	1.33	1.4	1.47	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
<b>0.93</b>	0.0	1.33	1.4	1.47	1.56	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
<b>0.91</b>	0.0	0.0	1.44	1.56	1.56	1.65	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
<b>0.90</b>	0.0	0.0	0.0	0.0	1.56	1.65	1.74	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
<b>0.88</b>	0.0	0.0	0.0	0.0	0.0	1.65	1.74	1.85	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
<b>0.86</b>	0.0	0.0	0.0	0.0	0.0	1.65	1.74	1.85	1.97	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	
<b>0.84</b>	0.0	0.0	0.0	0.0	0.0	0.0	1.74	1.85	1.97	2.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	
<b>0.81</b>	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.85	1.97	2.1	2.24	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	
<b>0.79</b>	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.97	2.1	2.24	2.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	
<b>0.76</b>	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.1	2.24	2.4	2.57	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	
<b>0.74</b>	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.1	2.24	2.4	2.57	2.76	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	
<b>0.71</b>	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.1	2.24	2.4	2.57	2.76	2.98	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	
<b>0.68</b>	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.1	2.24	2.4	2.57	2.76	2.98	3.22	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	
<b>0.64</b>	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.1	2.24	2.4	2.57	2.76	2.98	3.22	3.49	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	
<b>0.60</b>	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.1	2.24	2.4	2.57	2.76	2.98	3.22	3.49	3.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	
<b>0.56</b>	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.24	2.4	2.57	2.76	2.98	3.22	3.49	3.8	4.15	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	
<b>0.52</b>	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.24	2.4	2.57	2.76	2.98	3.22	3.49	3.8	4.15	4.56	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	
<b>0.46</b>	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.24	2.4	2.57	2.76	2.98	3.22	3.49	3.8	4.15	4.56	5.02	0.0	0.0	0.0	0.0	0.0	0.0	0	
<b>0.41</b>	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.24	2.4	2.57	2.76	2.98	3.22	3.49	3.8	4.15	4.56	5.02	5.56	0.0	0.0	0.0	0.0	0.0	0	
<b>0.35</b>	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.57	2.76	2.98	3.22	3.49	3.8	4.15	4.56	5.02	5.56	6.19	0.0	0.0	0.0	0	
<b>0.28</b>	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.57	2.76	2.98	3.22	3.49	3.8	4.15	4.56	5.02	5.56	6.19	6.93	0.0	0.0	0	
<b>0.20</b>	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.57	2.76	2.98	3.22	3.49	3.8	4.15	4.56	5.02	5.56	6.19	6.93	7.82	0.0	0	
<b>0.11</b>	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.57	2.76	2.98	3.22	3.49	3.8	4.15	4.56	5.02	5.56	6.19	6.93	7.82	8.89	0.0	
<b>0.01</b>	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.98	3.22	3.49	3.8	4.15	4.56	5.02	5.56	6.19	6.93	7.82	8.89	10.2	0	
<b>0.00</b>	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.98	3.22	3.49	3.8	4.15	4.56	5.02	5.56	6.19	6.93	7.82	8.89	10.2	15	

## C.2 Inference Process Visualization

图18和图19给出了一个完整推断过程的可视化。左半部分是系数矩阵表13的推断过程，右半部分是系数矩阵表14的推断过程。第一列是 Self Guidance 的结果，也是预测模型的图像信号输入，第二列 Uncondition 的模型输出，第三列是 Condition 的模型输出，第四列是 Classifier Free Guide 的结果。

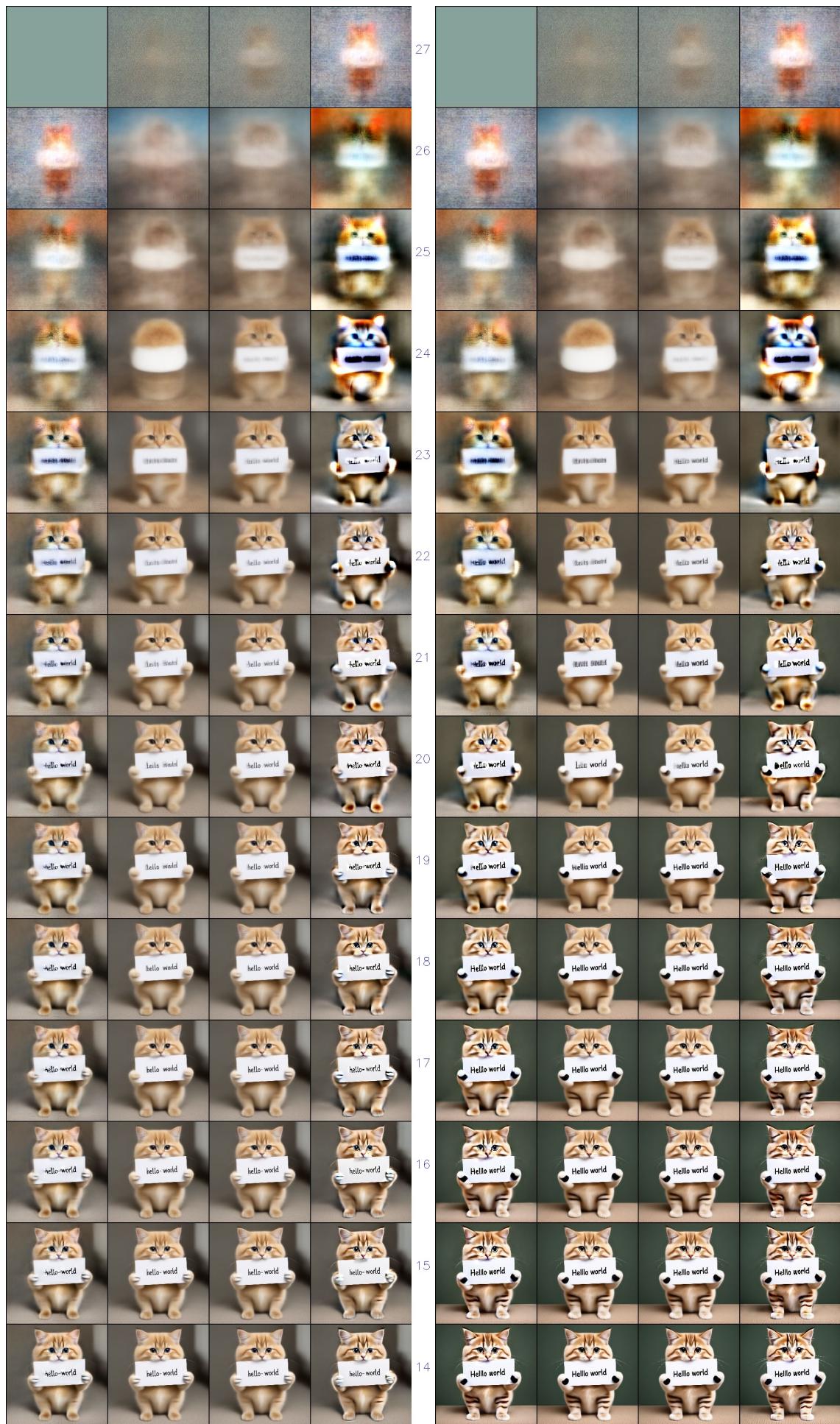


Figure 18: Inference process visualization: first half.

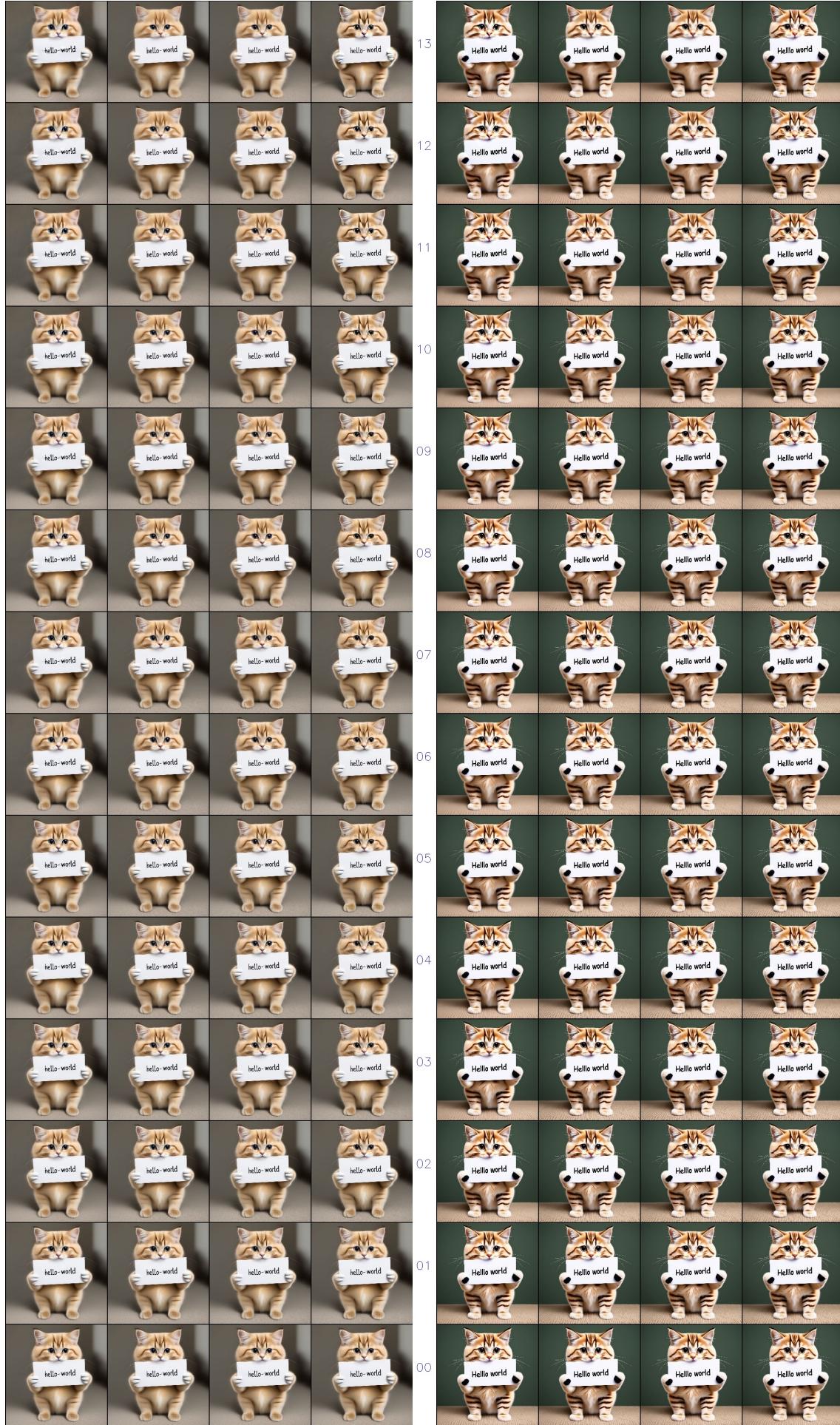


Figure 19: Inference process visualization: second half

## D Optimized Coefficient Matrix for Pretrained CIFAR10 Model

注意，为了直观理解每行各个系数的相对比例，表15的每行系数均是缩放后的结果，以确保对角元素等于1。使用时，需将每行归一化至每步相应的 Marginal Coefficient，例如，第一行需归一化至 0.118，第二行需归一化至 0.487。表16和17也是如此。

Table 15: optimized coefficient matrix for 5 step

time	<b>0.650</b>	<b>0.375</b>	<b>0.176</b>	<b>0.051</b>	<b>0.001</b>	marginal coeff
<b>0.650</b>	1	0	0	0	0	0.118
<b>0.375</b>	-0.291	1	0	0	0	0.487
<b>0.176</b>	0	0.133	1	0	0	0.85
<b>0.051</b>	0	0	-0.337	1	0	0.985
<b>0.001</b>	0	0	0	-0.583	1	1

Table 16: optimized coefficient matrix for 10 step

time	<b>0.816</b>	<b>0.650</b>	<b>0.503</b>	<b>0.375</b>	<b>0.266</b>	<b>0.176</b>	<b>0.104</b>	<b>0.051</b>	<b>0.017</b>	<b>0.001</b>	marginal coeff
<b>0.816</b>	1	0	0	0	0	0	0	0	0	0	0.035
<b>0.650</b>	0	1	0	0	0	0	0	0	0	0	0.118
<b>0.503</b>	0	-0.3	1	0	0	0	0	0	0	0	0.276
<b>0.375</b>	0	0.52	-0.3	1	0	0	0	0	0	0	0.487
<b>0.266</b>	0	0.2	0.4	-0.3	1	0	0	0	0	0	0.694
<b>0.176</b>	0	0	0.2	0.4	-0.2	1	0	0	0	0	0.85
<b>0.104</b>	0	0	0	0	0.35	-0.15	1	0	0	0	0.943
<b>0.051</b>	0	0	0	0	0	0.37	-0.2	1	0	0	0.985
<b>0.017</b>	0	0	0	0	0	0	0.1	-0.33	1	0	0.998
<b>0.001</b>	0	0	0	0	0	0	0	0.05	-0.34	1	1

Table 17: optimized coefficient matrix for 15 step

time	<b>0.875</b>	<b>0.758</b>	<b>0.650</b>	<b>0.550</b>	<b>0.459</b>	<b>0.375</b>	<b>0.300</b>	<b>0.234</b>	<b>0.176</b>	<b>0.126</b>	<b>0.084</b>	<b>0.051</b>	<b>0.026</b>	<b>0.009</b>	<b>0.001</b>	marginal coeff
<b>0.875</b>	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.021
<b>0.758</b>	0.24	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0.055
<b>0.650</b>	0.1	0.48	1	0	0	0	0	0	0	0	0	0	0	0	0	0.118
<b>0.550</b>	0	0.2	0.41	1	0	0	0	0	0	0	0	0	0	0	0	0.216
<b>0.459</b>	0	0	0.2	-0.2	1	0	0	0	0	0	0	0	0	0	0	0.343
<b>0.375</b>	0	0.3	-0.14	0.56	-0.77	1	0	0	0	0	0	0	0	0	0	0.487
<b>0.300</b>	0	0	0.23	-0.06	0.3	-0.85	1	0	0	0	0	0	0	0	0	0.629
<b>0.234</b>	0	0	0	0.26	-0.01	0.85	-0.86	1	0	0	0	0	0	0	0	0.753
<b>0.176</b>	0	0	0	0	0.25	0	0.82	-0.78	1	0	0	0	0	0	0	0.85
<b>0.126</b>	0	0	0	0	0	0.23	-0.02	0.9	-0.2	1	0	0	0	0	0	0.919
<b>0.084</b>	0	0	0	0	0	0	0.2	-0.04	0.7	-0.4	1	0	0	0	0	0.961
<b>0.051</b>	0	0	0	0	0	0	0.17	-0.07	0.62	-0.66	1	0	0	0	0	0.985
<b>0.026</b>	0	0	0	0	0	0	0	0.14	-0.09	0.57	-0.88	1	0	0	0	0.995
<b>0.009</b>	0	0	0	0	0	0	0	0	0.11	-0.11	0.31	-0.49	1	0	0	0.999
<b>0.001</b>	0	0	0	0	0	0	0	0	0	0.08	-0.11	0.24	-0.31	1	1	1