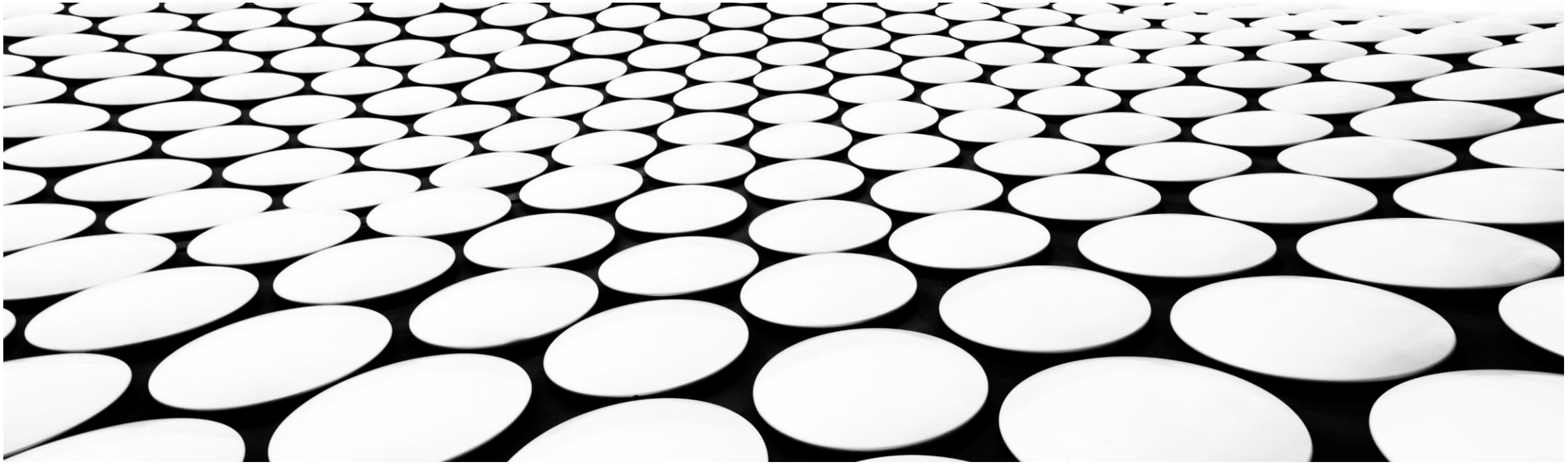
DIFFUSION的训练与GUIDED DIFFUSION



DDIM（DENOISING DIFFUSION IMPLICIT MODEL）

DDPM: $L_{t-1}^{\text{simple}} = \mathbb{E}_{\mathbf{x}_0,\epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\|\epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t\right)\|^2\right]$

DDPM其实仅仅依赖边缘分布$q(\mathbf{x}_t|\mathbf{x}_0)$

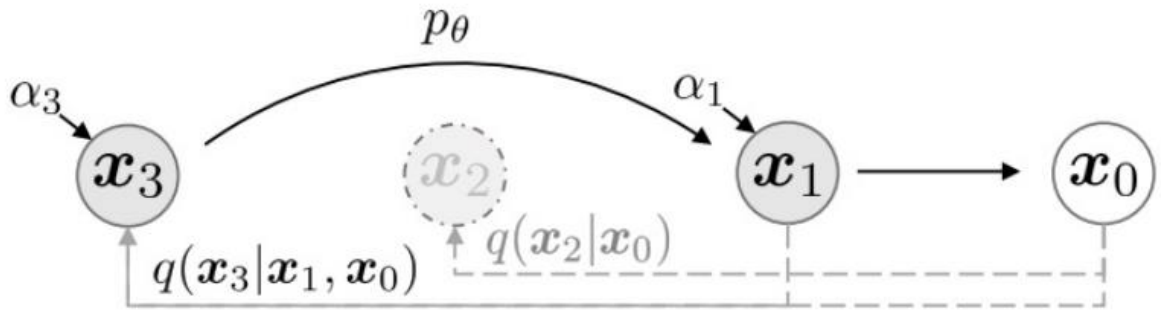
而并不是直接作用在联合分布$q(\mathbf{x}_{1:T}|\mathbf{x}_0)$

$$\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}}\underbrace{\left(\frac{\mathbf{x}_t - \sqrt{1-\alpha_t}\epsilon_\theta(\mathbf{x}_t, t)}{\sqrt{\alpha_t}}\right)}_{\text{predicted } \mathbf{x}_0} + \underbrace{\sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\epsilon_\theta(\mathbf{x}_t, t)}_{\text{direction pointing to } \mathbf{x}_t} + \underbrace{\sigma_t\epsilon_t}_{\text{random noise}}$$

$$\sigma_t^2 = \eta\cdot\tilde{\beta}_t = \eta\cdot\sqrt{(1-\alpha_{t-1})/(1-\alpha_t)}\sqrt{(1-\alpha_t/\alpha_{t-1})}$$

这里考虑两种情况，一是$\eta = 1$，此时$\sigma_t^2 = \tilde{\beta}_t$，此时生成过程就和DDPM一样了。另外一种情况是$\eta = 0$，这个时候生成过程就没有随机噪音了，是一个确定性的过程，论文将这种情况下的模型称为**DDIM（denoising diffusion implicit model）**，一旦最初的随机噪音$\mathbf{x}_T$确定了，那么DDIM的样本生成就变成了确定的过程。

DDIM加速过程——前向移动省略步数



那么生成过程也可以用这个子序列的反向马尔卡夫链来替代，由于S可以设置比原来的步数L要小，那么就可以加速生成过程。这里的生成过程变成：

$$\mathbf{x}_{\tau_{i-1}}=\sqrt{\alpha_{\tau_{i-1}}}\left(\frac{\mathbf{x}_{\tau_i}-\sqrt{1-\alpha_{\tau_i}}\epsilon_\theta(\mathbf{x}_{\tau_i},\tau_i)}{\sqrt{\alpha_{\tau_i}}}\right)+\sqrt{1-\alpha_{\tau_{i-1}}-\sigma_{\tau_i}^2}\cdot\epsilon_\theta(\mathbf{x}_{\tau_i},\tau_i)+\sigma_{\tau_i}\epsilon$$

其实上述的加速，我们是将前向过程按如下方式进行了分解：

$$q_{\sigma,\tau}(\mathbf{x}_{1:T}|\mathbf{x}_0)=q_{\sigma,\tau}(\mathbf{x}_T|\mathbf{x}_0)\prod_{i=1}^{S}q_\sigma(\mathbf{x}_{\tau_{i-1}}|\mathbf{x}_{\tau_i},\mathbf{x}_0)\prod_{t\in\bar{\tau}}q_{\sigma,\tau}(\mathbf{x}_t|\mathbf{x}_0)$$

其中$\bar{\tau}=\{1,\dots,T\}\backslash\tau$。这包含了两个图：其中一个就是由$\{\mathbf{x}_{\tau_i}\}_{i=1}^S$组成的马尔可夫链，另外一个是剩余的变量$\{\mathbf{x}_t\}_{t\in\bar{\tau}}$组成的星状图。同时生成过程，我们也只用马尔可夫链的那部分来生成：

$$p_\theta(\mathbf{x}_{0:T})=p(\mathbf{x}_T)\underbrace{\prod_{i=1}^{S}p_\theta(\mathbf{x}_{\tau_{i-1}}|\mathbf{x}_{\tau_i})}_{\text{use to produce sample}}\times\underbrace{\prod_{t\in\bar{\tau}}p_\theta(\mathbf{x}_0|\mathbf{x}_t)}_{\text{only for VLB}}$$

论文共设计了两种方法来采样子序列，分别是：

- **Linear**：采用线性的序列$\tau_i=\lfloor ci\rfloor$；
- **Quadratic**：采样二次方的序列$\tau_i=\lfloor ci^2\rfloor$；

C为定值

实验结果

下表为不同的η下以及不同采样步数下的对比结果，可以看到DDIM（$\eta = 0$）在较短的步数下就能得到比较好的效果，媲美DDPM（$\eta = 1$）的生成效果。如果S设置为50，那么相比原来的生成过程就可以加速20倍。

Table 1: CIFAR10 and CelebA image generation measured in FID. $\eta = 1.0$ and $\hat{\sigma}$ are cases of DDPM (although Ho et al. (2020) only considered $T = 1000$ steps, and $S < T$ can be seen as simulating DDPMs trained with S steps), and $\eta = 0.0$ indicates DDIM.

		CIFAR10 (32 × 32)					CelebA (64 × 64)				
	S	10	20	50	100	1000	10	20	50	100	1000
η	0.0	**13.36**	**6.84**	**4.67**	**4.16**	4.04	**17.33**	**13.73**	**9.17**	**6.53**	3.51
	0.2	14.04	7.11	4.77	4.25	4.09	17.66	14.11	9.51	6.79	3.64
	0.5	16.66	8.35	5.25	4.46	4.29	19.86	16.06	11.01	8.09	4.28
	1.0	41.07	18.36	8.01	5.78	4.73	33.12	26.03	18.48	13.93	5.98
$\hat{\sigma}$		367.43	133.37	32.72	9.99	**3.17**	299.71	183.83	71.71	45.20	**3.26**

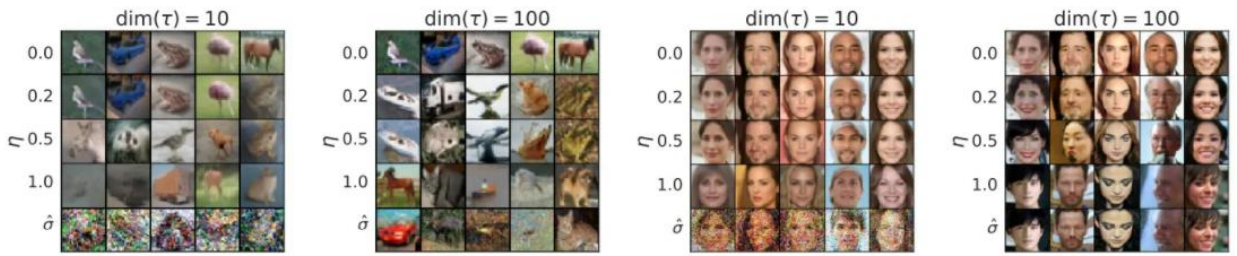


Figure 3: CIFAR10 and CelebA samples with $\dim(\tau) = 10$ and $\dim(\tau) = 100$.

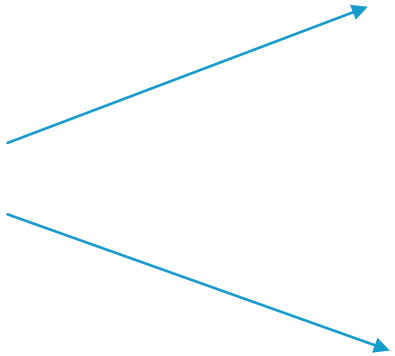
DIFFUSION的训练参数

```
modelConfig = {
    "state": "train", # or eval
    "epoch": 200,
    "batch_size": 80,
    "T": 1000,
    "channel": 128,
    "channel_mult": [1, 2, 3, 4],
    "attn": [2],
    "num_res_blocks": 2,
    "dropout": 0.15,
    "lr": 1e-4,
    "multiplier": 2.,
    "beta_1": 1e-4,
    "beta_T": 0.02,
    "img_size": 32,
    "grad_clip": 1.,
    "device": "cuda:0",
    "training_load_weight": None,
    "save_weight_dir": "./Checkpoints/",
    "test_load_weight": "ckpt_199_.pt",
    "sampled_dir": "./SampledImgs/",
    "sampledNoisyImgName": "NoisyNoGuidenceImgs.png",
    "sampledImgName": "SampledNoGuidenceImgs.png",
    "nrow": 8
    }
```

DIFFUSION的训练结果

GUIDED DIFFUSION（11 MAY 2021）

随机输入一张高斯噪声显然不能按照人的意愿生成我们想要的内容，因而需要额外的引导 guidance 以得到我们需要的图像。



一种想法是使用**外部模型**（分类器 or 广义的判别器）的输出作为引导条件来指导扩散模型的去噪过程，从而得到我们想要的输出

第二种想法：我们直接把我们想要的引导条件 condition 也作为**模型输入**的一部分，从而让扩散模型见到这个条件后就可以直接生成我们想要的内容。

CLASSIFIER GUIDANCE DIFFUSION MODEL

这种方法不用额外训练扩散模型，直接在原有训练好的扩散模型上，通过外部的分类器来引导生成期望的图像。唯一需要改动的地方其实只有 sampling 过程中的高斯采样的均值，也即采样过程中，期望噪声图像的采样中心越靠近判别器引导的条件越好。

Algorithm 1 Classifier guided diffusion sampling, given a diffusion model $(\mu_\theta(x_t), \Sigma_\theta(x_t))$, classifier $p_\phi(y|x_t)$, and gradient scale s.

Input: class label y, gradient scale s
$x_T \leftarrow$ sample from $\mathcal{N}(0, \mathbf{I})$
for all t from T to 1 **do**
  $\mu, \Sigma \leftarrow \mu_\theta(x_t), \Sigma_\theta(x_t)$
  $x_{t-1} \leftarrow$ sample from $\mathcal{N}(\mu + s\Sigma \nabla_{x_t} \log p_\phi(y|x_t), \Sigma)$
end for
return x_0

Algorithm 2 Classifier guided DDIM sampling, given a diffusion model $\epsilon_\theta(x_t)$, classifier $p_\phi(y|x_t)$, and gradient scale s.

Input: class label y, gradient scale s
$x_T \leftarrow$ sample from $\mathcal{N}(0, \mathbf{I})$
for all t from T to 1 **do**
  $\hat{\epsilon} \leftarrow \epsilon_\theta(x_t) - \sqrt{1-\bar{\alpha}_t} \nabla_{x_t} \log p_\phi(y|x_t)$
  $x_{t-1} \leftarrow \sqrt{\bar{\alpha}_{t-1}} \left(\frac{x_t - \sqrt{1-\bar{\alpha}_t}\hat{\epsilon}}{\sqrt{\bar{\alpha}_t}} \right) + \sqrt{1-\bar{\alpha}_{t-1}}\hat{\epsilon}$
end for
return x_0

上图总结了采样算法。Algorithm 1 和 Algorithm 2 其实是等价的（1 是直接预测均值和方差，2 是预测噪声的误差）。直接看 Algorithm 1 可知，实质上**改变的只有高斯分布的均值中心**，将扩散方向“引导”成我们想要的内容。具体而言，用分类模型 对生成的图片进行分类，得到预测分数与目标类别的交叉熵，将其对带噪图像求梯度用梯度引导下一步的生成采样。（实际使用的时候，需要把这个分类器也在带噪数据额外训练一下）

$$\mu, \Sigma \leftarrow \mu_\theta(x_t), \Sigma_\theta(x_t)$$

$$x_{t-1} \leftarrow \text{sample from } \mathcal{N}(\mu + s\Sigma g, \Sigma) \text{ , where } g = \nabla_{x_t} log(p_\phi(y|x_t))$$

这里 s 也是一个常量。即在每一步过程中，在计算高斯分布的均值时加上方差和分类梯度项的乘积。基于这样的改进，不需要重新训练扩散模型，只需要额外训练一个分类器，就能够有效地在添加类别引导。当然，这样的结构也存在一点小问题，就是会引入比较多的额外计算时间（每一步都要过分类模型并求梯度）。