PROGRESSIVE DISTILLATION FOR FAST SAMPLING OF DIFFUSION MODELS

A Preprint

Authors present a procedure to distill the behavior of a N-step DDIM sampler for a pretrained diffusion model into a new model with $\frac{N}{2}$ steps, with little degradation in sample quality. $x \sim p(x)$ - training data; latent variables: $z = \{z_t | t \in [0;1]\}$ and is specified by a noise schedule comprising differentiable functions a_t, σ_t such that $\lambda_t = \log \frac{a_t^2}{\sigma_t^2}$

Forward process is defined as $q(z|x) = \mathcal{N}(z_t, x; a_t \sigma^2 I)$; $q(z_t|z_s) = \mathcal{N}(z_t; \frac{a_t}{a_s} z_s; \sigma_{t|s}^2 I)$

where $0 < s < t \leq 1 \quad \sigma_{t|s}^2 = (1 - e^{\lambda_t - \lambda_s})\sigma_t^2$. $\hat{x_\theta}$ is trained; objective: $\mathbb{E}_{\epsilon;t}[w(\lambda t)||\hat{x}_\theta(z_t) - x||_2^2]$

To define the sampler, consider that the forward diffusion process admits a time-reversed description:

$$q(z_s \mid z_t, x) = \mathcal{N}(z_s; \tilde{\mu}_{s|t}(z_t, x), \tilde{\sigma}_{s|t}^2 I), \quad s < t, \tag{1}$$

where the mean and variance are given by:

$$\tilde{\mu}_{s|t}(z_t, x) = e^{\lambda_t - \lambda_s} \left(\frac{\alpha_s}{\alpha_t} \right) z_t + \left(1 - e^{\lambda_t - \lambda_s}\right) \alpha_s x, \tag{2}$$

$$\tilde{\sigma}_{s|t}^2 = \left(1 - e^{\lambda_t - \lambda_s}\right) \sigma_s^2. \tag{3}$$

This reversed formulation enables the construction of an ancestral sampler. Starting from $z_1 \sim \mathcal{N}(0, I)$, the sampler iteratively generates:

$$z_s = \tilde{\mu}_{s|t}(z_t, \hat{x}_\theta(z_t)) + \sqrt{{\tilde{\sigma}_{s|t}^2}^{1-\gamma} \cdot {\sigma_{t|s}^2}^{\gamma}} \cdot \varepsilon, \tag{4}$$

or equivalently:

$$z_s = e^{\lambda_t - \lambda_s} \left(\frac{\alpha_s}{\alpha_t} \right) z_t + \left(1 - e^{\lambda_t - \lambda_s}\right) \alpha_s \hat{x}_\theta(z_t) + \sqrt{{\tilde{\sigma}_{s|t}^2}^{1-\gamma} \cdot {\sigma_{t|s}^2}^{\gamma}} \cdot \varepsilon, \tag{5}$$

where $\varepsilon \sim \mathcal{N}(0, I)$ is standard Gaussian noise and γ is a hyperparameter regulating the noise level during sampling.

An alternative approach maps initial noise $z_1 \sim \mathcal{N}(0, I)$ deterministically to a sample x by solving the probability flow ODE:

$$\frac{dz_t}{dt} = f(z_t, t) - \frac{1}{2} g^2(t) \nabla_z \log \hat{p}_\theta(z_t), \tag{6}$$

with the score estimate:

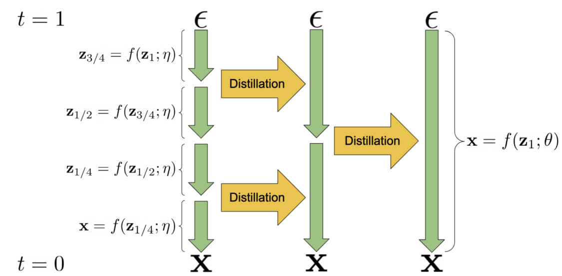

Figure 1: A visualization of two iterations of our proposed *progressive distillation* algorithm. A sampler $f(\mathbf{z}; \eta)$, mapping random noise ϵ to samples $\mathbf{x}$ in 4 deterministic steps, is distilled into a new sampler $f(\mathbf{z}; \theta)$ taking only a single step. The original sampler is derived by approximately integrating the *probability flow ODE* for a learned diffusion model, and distillation can thus be understood as learning to integrate in fewer steps, or *amortizing* this integration into the new sampler.

$$\nabla_z \log \hat{p}_\theta(z_t) = \frac{\alpha_t \hat{x}_\theta(z_t) - z_t}{\sigma_t^2}, \tag{7}$$

and dynamics defined by:

$$f(z_t, t) = \frac{d \log \alpha_t}{dt} \cdot z_t, \tag{8}$$

$$g^2(t) = \frac{d\sigma_t^2}{dt} - 2\frac{d \log \alpha_t}{dt} \cdot \sigma_t^2. \tag{9}$$

Since $\hat{x}_\theta(z_t)$ is parameterized by a neural network, the ODE above constitutes a special case of a neural ODE, or equivalently, a continuous normalizing flow. Numerical integration of this ODE can be performed using methods such as Euler or Runge–Kutta.

The DDIM sampling scheme can be interpreted as a specific discretization of this ODE. Its update rule is given by:

$$z_s = \alpha_s \hat{x}_\theta(z_t) + \sigma_s \cdot \frac{z_t - \alpha_t \hat{x}_\theta(z_t)}{\sigma_t}, \tag{10}$$

or, in exponential form:

$$z_s = e^{(\lambda_t - \lambda_s)/2} \left(\frac{\alpha_s}{\alpha_t}\right) z_t + \left(1 - e^{(\lambda_t - \lambda_s)/2}\right) \alpha_s \hat{x}_\theta(z_t). \tag{11}$$

Empirical observations indicate that this rule often outperforms standard ODE solvers in practice.

Under mild smoothness assumptions on $\hat{x}_\theta(z_t)$, the numerical integration error vanishes in the limit of infinitely many steps, i.e., as $N \to \infty$. However, in practical settings, hundreds or thousands of steps are typically required to obtain high-quality samples, which is computationally expensive. To mitigate this issue, a distillation technique is proposed to approximate accurate but slow solvers with faster models, while maintaining sample quality.

The Trade-off: Using DDIM with a huge number of steps gives you very high-quality images because you're solving the ODE very accurately. But this is incredibly slow. The Goal: The authors want to create a model that gets the same high-quality results but in far fewer steps.

The main difference from in this type of distillation is moving not towards image, but towards $\hat{x}$ that makes a single student DDIM step match 2 teacher DDIM steps

Algorithm 1 Progressive Distillation

Require: Trained teacher model $\hat{x}_\eta(z_t)$; dataset $\mathcal{D}$; loss weight function $w(\cdot)$; initial number of student sampling steps N

for each of K distillation iterations **do**
  Initialize student: $\theta \leftarrow \eta$
  while not converged **do**
    Sample data: $x \sim \mathcal{D}$
    Sample timestep index: $i \sim \text{Cat}([1, 2, \dots, N])$
    Set timestep: $t \leftarrow i/N$
    Sample noise: $\varepsilon \sim \mathcal{N}(0, I)$
    Encode: $z_t \leftarrow \alpha_t x + \sigma_t \varepsilon$
                                    ▷ Two DDIM steps with teacher
    $t' \leftarrow t - 0.5/N$
    $t'' \leftarrow t - 1/N$
    $z_{t'} \leftarrow \alpha_{t'} \hat{x}_\eta(z_t) + \sigma_{t'} \cdot \frac{z_t - \alpha_t \hat{x}_\eta(z_t)}{\sigma_t}$
    $z_{t''} \leftarrow \alpha_{t''} \hat{x}_\eta(z_{t'}) + \sigma_{t''} \cdot \frac{z_{t'} - \alpha_{t'} \hat{x}_\eta(z_{t'})}{\sigma_{t'}}$
                                    ▷ Construct teacher prediction target
    $\tilde{x} \leftarrow \frac{z_{t''} - (\sigma_{t''}/\sigma_t) z_t}{\alpha_{t''} - (\sigma_{t''}/\sigma_t)\alpha_t}$
    $\lambda_t \leftarrow \log(\alpha_t^2/\sigma_t^2)$
                                    ▷ Optimize student to match teacher
    $\mathcal{L}_\theta \leftarrow w(\lambda_t) \cdot \|\tilde{x} - \hat{x}_\theta(z_t)\|_2^2$
    $\theta \leftarrow \theta - \gamma \nabla_\theta \mathcal{L}_\theta$
  end while
  Update teacher: $\eta \leftarrow \theta$
  Halve sampling steps: $N \leftarrow N/2$
end for