Pseudocode of ADDT

$\mathbf{x}_0$ is image from training dataset, $\mathbf{y}$ is the class label of the image, C is the classifier, P is one-step diffusion reverse process and θ is it's parameter, $\mathrm{L}$is CrossEntropy Loss.
Here, we take DDPM for example, The training process is as follows:

- for $\mathbf{x}_0$, $\mathbf{y}$ in the training dataset do:
  - $t \sim \mathrm{Uniform}(\{1, ..., T\})$
  - $\lambda_t = \mathrm{clip}(\gamma_t \lambda_{\mathrm{unit}}, \lambda_{\mathrm{min}}, \lambda_{\mathrm{max}})$, where $\gamma_t = \dfrac{\sqrt{\overline{\alpha}_t}}{\sqrt{1-\overline{\alpha}_t}}$
  - $\delta = 0$
  - for 1 to $\mathrm{ADDT}_{\mathrm{iterations}}$ do:
    - $\epsilon \sim \mathcal{N}(0, I)$
    - $\epsilon_\delta = \mathrm{RBGM}(\delta)$
    - $\mathbf{x}_t = \sqrt{\overline{\alpha}_t}\,\mathbf{x}_0 + \lambda_t \sqrt{1-\overline{\alpha}_t}\,\epsilon + \sqrt{1-\lambda_t^2}\,\sqrt{1-\overline{\alpha}_t}\,\epsilon_\delta$
    - $\delta += \nabla_{\epsilon_\delta} L(C(P(\mathbf{x}_t, t), \mathbf{y}))$

    end for
  - $\epsilon \sim \mathcal{N}(0, I)$
  - $\epsilon_\delta = \mathrm{RBGM}(\delta)$
  - $\mathbf{x}_t = \sqrt{\overline{\alpha}_t}\,\mathbf{x}_0 + \lambda_t \sqrt{1-\overline{\alpha}_t}\,\epsilon + \sqrt{1-\lambda_t^2}\,\sqrt{1-\overline{\alpha}_t}\,\epsilon_\delta$
  - Take a gradient descent step on:
    - $\nabla_\theta \| \dfrac{\sqrt{\overline{\alpha}_t}}{\sqrt{1-\overline{\alpha}_t}} (\mathbf{x}_0 - P(\mathbf{x}_t, t)) \|_2^2$

  end for

In DDPM, the unet $\boldsymbol{\epsilon_\theta}$ predicts the Gaussian noise added to the image, adopting equation (3) in the paper, we have $P(\mathbf{x}_t, t) = \left(\mathbf{x}_t - \sqrt{1-\overline{\alpha}_t}\,\boldsymbol{\epsilon_\theta}(\mathbf{x}_t, t)\right) / \sqrt{\overline{\alpha}_t}$