A. Investigation of Computational Overhead

In addition to the evaluation of sampling qualities, we also examined the computational overhead introduced by the extrapolation operations in LA-DPMs. The minibatch size in the sampling procedure was set to 500, and the running time was measured over a windows machine with a 1080ti GPU. Table 2 displays the time complexities of different generative models over CIFAR10. It is clear from the table that the computational overhead of the extrapolation operations in LA-DPMs is negligible. This is because the extrapolation operations are linear and no additional DNN models are introduced to assist the operations.

Table 2. Comparison of computational costs (measured in units of seconds per minibatch) for CIFAR10.

Timesteps	10	25	50	100	200	1000
NPR-DDPM	14.9	36.4	70.9	139.6	278.1	1388.5
LA-NPR-DDPM	15.3	36.9	71.6	139.8	279.9	1389.9
SN-DDPM	14.9	36.5	71.0	140.1	278.0	1387.9
LA-SN-DDPM	15.3	37.5	71.9	140.8	278.4	1390.7
NPR-DDIM	14.9	36.2	70.7	139.7	270.2	1385.4
LA-NPR-DDIM	15.4	36.3	71.5	140.3	271.1	1388.5
SN-DDIM	15.4	36.6	71.1	136.2	279.1	1386.2
LA-SN-DDIM	15.6	37.0	71.3	139.3	280.9	1391.3

B. Additional Ablation Study of LA-DPMs

We have conducted additional ablation studies for LA-DPMs over both CIFAR10 and ImageNet64. Our main objective is to show that the optimal setup for the parameter λ is different for different timesteps.

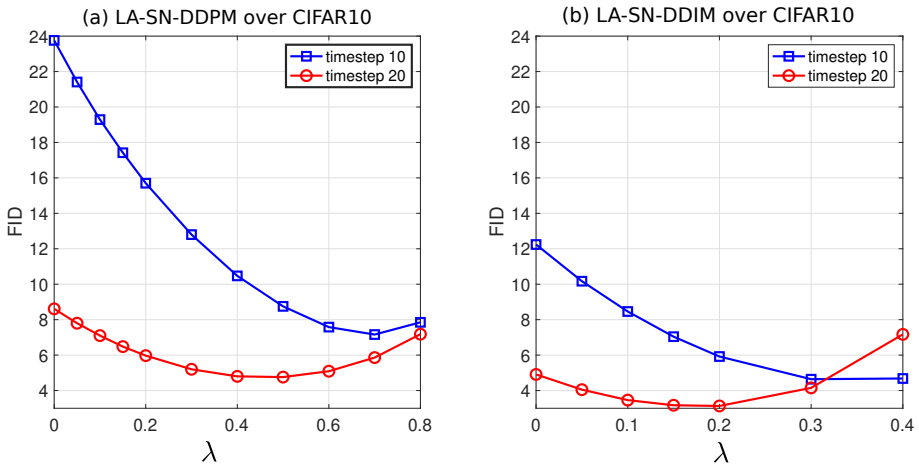


Figure 5. FID scores versus λ values for LA-SN-DDPM and LA-SN-DDIM over CIFAR10. When $\lambda = 0$, LA-SN-DDPM reduces to SN-DDPM and LA-SN-DDIM reduces to SN-DDIM.

It is seen from both Fig. 5 and 6 that as λ increases, the FID score first decreases then increases. That is, it is preferable to select a proper nonzero λ to achieve small FID scores. Furthermore, as the timestep increases from 10 to 20, the optimal value for λ decreases. In other words, large λ values are preferable when the timestep for sampling is small. This also explains why the setup $\lambda = 0.1$ leads to higher FID scores for ImageNet64 for large timesteps (e.g., 200, 1000) in Table 1.

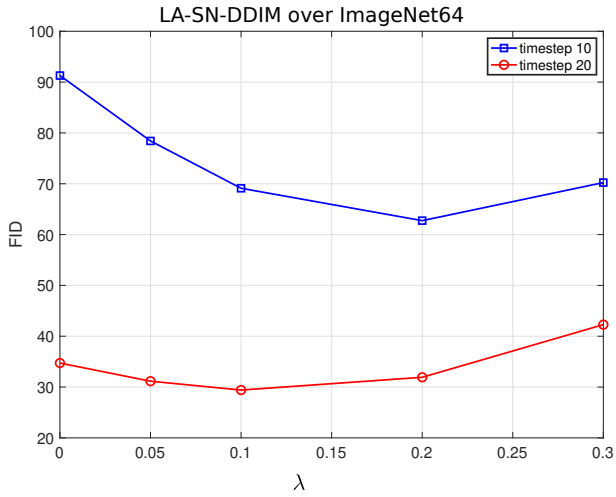


Figure 6. FID scores versus λ values for LA-SN-DDIM over ImageNet64. When $\lambda = 0$, LA-SN-DDIM reduces to SN-DDIM.

C. Lookahead High-Order DPM Solvers and Performance Comparison

C.1. LA-DPM-Solver-2

The update expression for DPM-Solver-2 takes the form of (see [14])

$$\begin{cases} t_{i-\frac{1}{2}} = t_\lambda(\frac{\lambda_{t_{i-1}}+\lambda_{t_i}}{2}) \\ \boldsymbol{z}_{i-\frac{1}{2}} = \frac{\alpha_{i-\frac{1}{2}}}{\alpha_i}\boldsymbol{z}_i - \sigma_{i-\frac{1}{2}}(e^{\frac{h_i}{2}}-1)\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i) \\ \boldsymbol{z}_{i-1} = \frac{\alpha_{i-1}}{\alpha_i}\boldsymbol{z}_i - \sigma_{i-1}(e^{h_i}-1)\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i) - \sigma_{i-1}(e^{h_i}-1)\left[\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}\left(\boldsymbol{z}_{i-\frac{1}{2}}, i-\frac{1}{2}\right) - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i)\right] \end{cases}, \tag{34}$$

where $\lambda_t = \log(\alpha_t/\sigma_t)$ is a strictly decreasing function and $t_\lambda(\cdot)$ is the reverse function of λ_t, and $h_i = \lambda_{t_{i-1}} - \lambda_{t_i}$. The expression for $\boldsymbol{z}_{i-\frac{1}{2}}$ in (34) can be simplified to be

$$\begin{aligned} \boldsymbol{z}_{i-\frac{1}{2}} &= \frac{\alpha_{i-\frac{1}{2}}}{\alpha_i}\boldsymbol{z}_i - \sigma_{i-\frac{1}{2}}(e^{\frac{h_i}{2}}-1)\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i) \\ &\left[\lambda_{t_{i-\frac{1}{2}}} = \frac{\lambda_{t_{i-1}}+\lambda_{t_i}}{2}, \text{which is obtained from definition of } t_{i-\frac{1}{2}} \text{ in (34)}\right] \\ &= \frac{\alpha_{i-\frac{1}{2}}}{\alpha_i}\boldsymbol{z}_i - \sigma_{i-\frac{1}{2}}\left(e^{\left(\lambda_{t_{i-\frac{1}{2}}} - \lambda_{t_i}\right)} - 1\right)\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i) \\ &\left[\lambda_{t_{i-\frac{1}{2}}} = \frac{\lambda_{t_{i-1}}+\lambda_{t_i}}{2} = \log(\alpha_{i-\frac{1}{2}}/\sigma_{i-\frac{1}{2}}),\ \ \lambda_{t_i} = \log(\alpha_i/\sigma_i)\right] \\ &= \frac{\alpha_{i-\frac{1}{2}}}{\alpha_i}\boldsymbol{z}_i - \sigma_{i-\frac{1}{2}}\left(\frac{\alpha_{i-\frac{1}{2}}}{\sigma_{i-\frac{1}{2}}}\frac{\sigma_i}{\alpha_i} - 1\right)\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i) \\ &= \alpha_{i-\frac{1}{2}}\underbrace{\left(\frac{\boldsymbol{z}_i}{\alpha_i} - \frac{\sigma_i}{\alpha_i}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i)\right)}_{\hat{\boldsymbol{x}}(\boldsymbol{z}_i, \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i))} + \sigma_{i-\frac{1}{2}}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i) \\ &\left[\text{For variance preserving process: } \sigma_{i-\frac{1}{2}} = \sqrt{1-\alpha^2_{i-\frac{1}{2}}}\right] \\ &= \alpha_{i-\frac{1}{2}}\hat{\boldsymbol{x}}(\boldsymbol{z}_i, \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i)) + \sqrt{1-\alpha^2_{i-\frac{1}{2}}}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i). \end{aligned} \tag{35}$$

LA-DPM-Solver-2 is designed by simply replacing $\hat{\boldsymbol{x}}(\boldsymbol{z}_i, \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i))$ in (35) with an extrapolation, given by

$$\boldsymbol{z}_{i-1} = \alpha_{i-\frac{1}{2}} \left[(1+\lambda_i)\hat{\boldsymbol{x}}(\boldsymbol{z}_i, \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i)) - \lambda_i \hat{\boldsymbol{x}} \left(\boldsymbol{z}_{i+\frac{1}{2}}, \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}} \left(\boldsymbol{z}_{i+\frac{1}{2}}, i+\frac{1}{2} \right) \right) \right] + \sqrt{1-\alpha^2_{i-\frac{1}{2}}} \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i). \tag{36}$$

The other quantities in LA-DPM-Solver-2 are computed in the same manner as DPM-Solver-2.

C.2. LA-DPM-Solver-3

We first present the update expressions for DPM-Solver-3 from [14].

$$\begin{cases} t_{i-\frac{1}{3}} = t_\lambda(\frac{\lambda_{t_{i-1}}+2\lambda_{t_i}}{3}) \\ t_{i-\frac{2}{3}} = t_\lambda(\frac{2\lambda_{t_{i-1}}+\lambda_{t_i}}{3}) \\ \boldsymbol{z}_{i-\frac{1}{3}} = \frac{\alpha_{i-\frac{1}{3}}}{\alpha_i}\boldsymbol{z}_i - \sigma_{i-\frac{1}{3}}(e^{\frac{h_i}{3}}-1)\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i) \\ \boldsymbol{r}_{i-\frac{1}{3}} = \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_{i-\frac{1}{3}}, i-\frac{1}{3}) - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i) \\ \boldsymbol{z}_{i-\frac{2}{3}} = \frac{\alpha_{i-\frac{2}{3}}}{\alpha_i}\boldsymbol{z}_i - \sigma_{i-\frac{2}{3}}(e^{\frac{2h_i}{3}}-1)\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i) - 2\sigma_{i-\frac{2}{3}}\left(\frac{e^{2h_i/3}-1}{(2h_i)/3}-1\right)\boldsymbol{r}_{i-\frac{1}{3}} \\ \boldsymbol{r}_{i-\frac{2}{3}} = \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_{i-\frac{2}{3}}, i-\frac{2}{3}) - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i) \\ \boldsymbol{z}_{i-1} = \frac{\alpha_{i-1}}{\alpha_i}\boldsymbol{z}_i - \sigma_{i-1}(e^{h_i}-1)\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i) - \frac{3\sigma_{i-1}}{2}\left(\frac{e^{h_i}-1}{h_i}-1\right)\boldsymbol{r}_{i-\frac{2}{3}}, \end{cases} \tag{37}$$

where $\lambda_t = \log(\alpha_t/\sigma_t)$ is a strictly decreasing function and $t_\lambda(\cdot)$ is the reverse function of λ_t, and $h_i = \lambda_{t_{i-1}} - \lambda_{t_i}$. The two timestep $t_{i-\frac{1}{3}}$ and $t_{i-\frac{2}{3}}$ are in between t_i and t_{i-1}. It clear from (37) that the computation of $\boldsymbol{z}_{i-1}$ involves a linear combination of $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i)$ and the difference vector $\boldsymbol{r}_{i-\frac{2}{3}} = \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_{i-\frac{2}{3}}, i-\frac{2}{3}) - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i)$.

Next, we study the update expression for $\boldsymbol{z}_{i-\frac{1}{3}}$ in (37), which can be reformulated as

$$\begin{aligned} \boldsymbol{z}_{i-\frac{1}{3}} &= \frac{\alpha_{i-\frac{1}{3}}}{\alpha_i}\boldsymbol{z}_i - \sigma_{i-\frac{1}{3}}(e^{\frac{h_i}{3}}-1)\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i) \\ &\left[\lambda_{t_{i-\frac{1}{3}}} = \frac{\lambda_{t_{i-1}}+2\lambda_{t_i}}{3}, \text{which is obtained from definition of } t_{i-\frac{1}{3}} \text{ in (37)}\right] \\ &= \frac{\alpha_{i-\frac{1}{3}}}{\alpha_i}\boldsymbol{z}_i - \sigma_{i-\frac{1}{3}}\left(e^{\left(\lambda_{t_{i-\frac{1}{3}}}-\lambda_{t_i}\right)}-1\right)\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i) \\ &\left[\lambda_{t_{i-\frac{1}{3}}} = \frac{\lambda_{t_{i-1}}+2\lambda_{t_i}}{3} = \log(\alpha_{i-\frac{1}{3}}/\sigma_{i-\frac{1}{3}}),\ \ \lambda_{t_i} = \log(\alpha_i/\sigma_i)\right] \\ &= \frac{\alpha_{i-\frac{1}{3}}}{\alpha_i}\boldsymbol{z}_i - \sigma_{i-\frac{1}{3}}\left(\frac{\alpha_{i-\frac{1}{3}}}{\sigma_{i-\frac{1}{3}}}\frac{\sigma_i}{\alpha_i}-1\right)\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i) \\ &= \alpha_{i-\frac{1}{3}}\underbrace{\left(\frac{\boldsymbol{z}_i}{\alpha_i} - \frac{\sigma_i}{\alpha_i}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i)\right)}_{\hat{\boldsymbol{x}}(\boldsymbol{z}_i, \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i))} + \sigma_{i-\frac{1}{3}}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i) \\ &\left[\text{For variance preserving process: } \sigma_{i-\frac{1}{3}} = \sqrt{1-\alpha^2_{i-\frac{1}{3}}}\right] \\ &= \alpha_{i-\frac{1}{3}}\hat{\boldsymbol{x}}(\boldsymbol{z}_i, \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i)) + \sqrt{1-\alpha^2_{i-\frac{1}{3}}}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i). \end{aligned} \tag{38}$$

To obtain the update expressions of LA-PDM-Solver-3, we modify (38) to be

$$\boldsymbol{z}_{i-\frac{1}{3}} = \alpha_{i-\frac{1}{3}} \left[(1+\lambda_i)\hat{\boldsymbol{x}}(\boldsymbol{z}_i, \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i)) - \lambda_i \hat{\boldsymbol{x}}(\boldsymbol{z}_{i+\frac{1}{3}}, \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_{i+\frac{1}{3}}, i+\frac{1}{3})) \right] + \sqrt{1-\alpha^2_{i-\frac{1}{3}}} \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i). \tag{39}$$

The computation for other quantities in LA-DPM-Solver-3 is the same as in DPM-Solver-3.

C.3. Evaluation of lookahead high-order DPM-Solvers

Experimental setups: In this experiment, we took two high-order DPM-solvers from [14] as two reference methods, which are DPM-Solver-2 and DPM-Solver-3. The two solvers essentially conduct extrapolation on the predicted Gaussian noises to improve the sampling quality. Our objective is to find out if their sampling quality can be further improved by performing additional extrapolation on the estimates of $\boldsymbol{x}$.

We utilized the same pre-trained model over CIFAR10 for evaluating tAB-DEIS and LA-tAB-DEIS in Subsection 5.3 (see Table 3). It is noted that the two high-order solvers in [14] were designed to work under a small number of timesteps (below 50 in the experiment of [14]). Therefore, in our experiment, the tested sampling steps are in the range of $[10, 40]$. In our improved methods, the strengths of the extrapolations were set to $\lambda_i = 0.1$, $i < N$.

Performance comparison: Fig. 7 summarises the FID scores versus timesteps. It is clear that even for high-order DPM-Solvers, the additional extrapolation on estimates of $\boldsymbol{x}$ helps to achieve lower FID scores. We can also conclude from the figure that DPM-Solver-3 outperforms DPM-Solver-2. This might be because DPM-Solver-3 manages to approximate the integration of the ODE (19) more accurately than DPM-Solver-2.

We note that Fig. 7 and Fig. 3 are based on the same pre-trained model for CIFAR10. By inspection of the FID scores in the two figures, it is clear that tAB-DEIS (order 3) performs better than DPM-Solver-3 for this particular pre-trained model. It is interesting from Fig. 3 that LA-tAB-DEIS outperforms tAB-DEIS significantly while performance gain of LA-DPM-Solver-3 over DPM-Solver-3 is moderate. The above results demonstrate that the performance gain of our lookahead technique depends on the original sampling method.

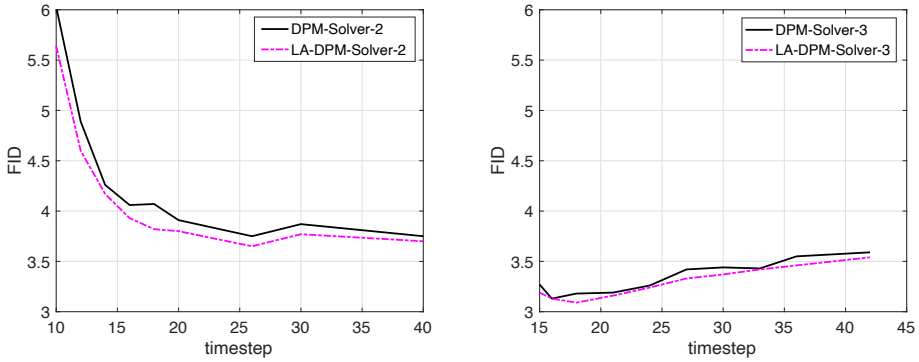


Figure 7. Performance of DPM-Solvers and LA-DPM-Solvers for CIFAR10.

D. Design of LA-S-PNDM

We summarize the sampling procedure of LA-S-PNDM in Alg. 2. The only difference between LA-S-PNDM and S-PNDM is the computation of $\boldsymbol{z}_{i-1}$ for $i = N-1, \ldots, 1$. It is seen from Alg. 2 that an additional extrapolation is introduced in terms of the estimates $\hat{\boldsymbol{x}}_{[i:i+1]}$ and $\tilde{\boldsymbol{x}}_{[i+1:i+2]}$ of the original data sample $\boldsymbol{x}$. The strengths of the extrapolations are parameterized by $\{\lambda_i\}_{i=1}^{N-1}$. When $\lambda_i = 0$ for all i, LA-S-PNDM reduces to S-PNDM.

From Alg. 2, we observe that the method S-PNDM or LA-S-PNDM exploits 2nd order polynomial of the estimated Gaussian noises $\{\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_{i+j}, i+j)\}_{j=0}^{1}$ in estimation of $\boldsymbol{z}_{i-1}$ at timestep i. The polynomial coefficients are computed differently for $i = N$ and $i < N$.

Algorithm 2 Sampling of LA-S-PNDM

Input: $\boldsymbol{z}_N \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, $\{1 > \lambda_i \geq 0\}_{i=1}^{N-1}$
for $i = N$ **do**

$$(a)\begin{cases} \boldsymbol{z}_{i-1} = \frac{\alpha_{i-1}}{\alpha_i}\left(\boldsymbol{z}_i - \sqrt{1-\alpha_i^2}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i)\right) + \sqrt{1-\alpha_{i-1}^2}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i) \\ \hat{\boldsymbol{\epsilon}}_{[i-1:i]} = \frac{1}{2}(\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i) + \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_{i-1}, i-1)) \\ \hat{\boldsymbol{x}}_i = (\boldsymbol{z}_i - \sqrt{1-\alpha_i^2}\hat{\boldsymbol{\epsilon}}_{[i-1:i]})/\alpha_i \\ \boldsymbol{z}_{i-1} = \alpha_{i-1}\hat{\boldsymbol{x}}_i + \sqrt{1-\alpha_{i-1}^2}\hat{\boldsymbol{\epsilon}}_{[i-1:i]} \end{cases}$$

end for
Denote $\tilde{\boldsymbol{x}}_{[N:N+1]} = \hat{\boldsymbol{x}}_N$
for $i = N-1, \ldots, 1$ **do**

$$(b)\begin{cases} \tilde{\boldsymbol{\epsilon}}_{[i:i+1]} = \frac{1}{2}(3\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_i, i) - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_{i+1}, i+1)) \\ \hat{\boldsymbol{x}}_{[i:i+1]} = (\boldsymbol{z}_i - \sqrt{1-\alpha_i^2}\tilde{\boldsymbol{\epsilon}}_{[i:i+1]})/\alpha_i \\ \boldsymbol{z}_{i-1} = \alpha_{i-1}\left((1+\lambda_i)\hat{\boldsymbol{x}}_{[i:i+1]} - \lambda_i\tilde{\boldsymbol{x}}_{[i+1:i+2]}\right) + \sqrt{1-\alpha_{i-1}^2}\tilde{\boldsymbol{\epsilon}}_{[i:i+1]} \\ \tilde{\boldsymbol{x}}[i:i+1] = (1+\lambda_i)\hat{\boldsymbol{x}}_{[i:i+1]} - \lambda_i\tilde{\boldsymbol{x}}_{[i+1:i+2]} \end{cases}$$

end for
output: $\boldsymbol{z}_0$

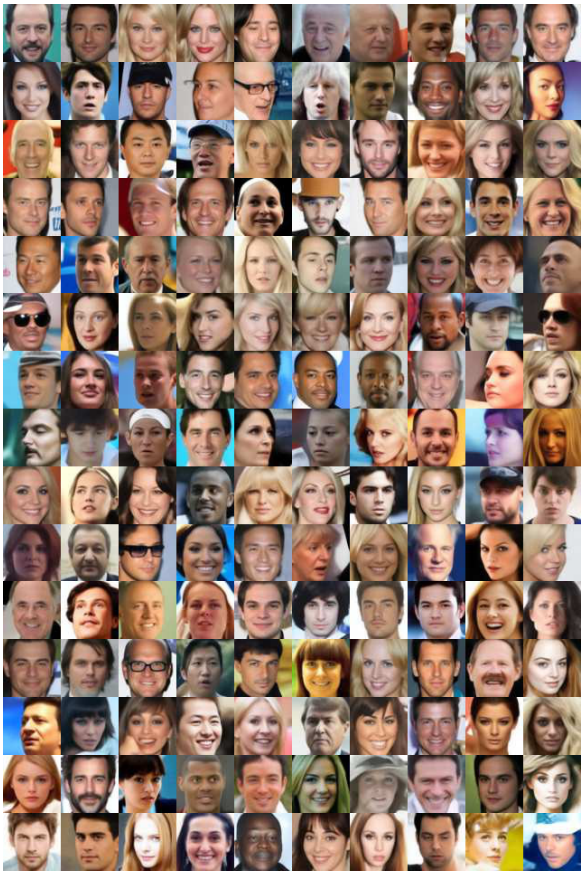
* The update for $\boldsymbol{z}_{N-1}$ in (a) is referred to as pseudo improved Euler step in [13].
* The update for $\boldsymbol{z}_{i-1}$ in (b) is referred to as pseudo linear multi step in [13].
* LA-S-PNDM reduces to S-PNDM when $\{\lambda_i = 0\}_{i=N-1}^{1}$.

E. Tested Pre-trained Models in Experiments

Table 3. sampling methods and the corresponding pre-trained models

sampling methods	model name
Fig. 3 for tAB-DEIS and LA-tAB-DEIS Fig. 7 for DPM-Solvers and LA-DPM-Solvers	cifar10_ddpmpp_deep_continuous/checkpoint_8.pth (from https://github.com/yang-song/score_sde)
Fig. 4 for S-PNDM and LA-S-PNDM	1.ddim_cifar10.ckpt 2.ddim_celeba.ckpt 3.ddim_lsun_bedroom.ckpt 4.ddim_lsun_church.ckpt (from https://github.com/luping-liu/PNDM)

(a) LA-tAB-DEIS (order 3) on CIFAR10 (timestep 10)



(b) LA-S-PNDM on CelebA64 (timestep 5)

Figure 8. Generated images with LA-tAB-DEIS and LA-S-PNDM

(a) LA-S-PNDM on bedroom (timestep 10)

(b) LA-S-PNDM on church (timestep 10)

Figure 9. Generated images with LA-S-PNDM