Artificial Intelligence Review (2024) 57:161
https://doi.org/10.1007/s10462-024-10800-8

Lightweight diffusion models: a survey

Wei Song[1,2,3,4] · Wen Ma[1] · Ming Zhang[1] · Yanghao Zhang[1] · Xiaobing Zhao[1,2,4]

Accepted: 16 May 2024

Abstract

Diffusion models (DMs) are a type of potential generative models, which have achieved better effects in many fields than traditional methods. DMs consist of two main processes: one is the forward process of gradually adding noise to the original data until pure Gaussian noise; the other is the reverse process of gradually removing noise to generate samples conforming to the target distribution. DMs optimize the application results through the iterative noise processing process. However, this greatly increases the computational and storage costs in the training and inference stages, limiting the wide application of DMs. Therefore, how to effectively reduce the resource consumption of using DMs while giving full play to their good performance has become a valuable and necessary research problem. At present, some research has been devoted to lightweight DMs to solve this problem, but there has been no survey in this area. This paper focuses on lightweight DMs methods in the field of image processing, classifies them according to their processing ideas. Finally, the development prospect of future work is analyzed and discussed. It is hoped that this paper can provide other researchers with strategic ideas to reduce the resource consumption of DMs, thereby promoting the further development of this research direction and providing available models for wider applications.

Keywords Diffusion models · Lightweight · Image processing · Generative models

1 Introduction

Diffusion models (DMs), a class of score-based generative models, have attracted increasing attention in recent years due to their powerful performance. Compared with generative adversarial networks (GANs) (Goodfellow et al. 2014; Gui et al. 2023), DMs can

✉ Wei Song
songwei@muc.edu.cn

[1] School of Information Engineering, Minzu University of China, Beijing 100081, China

[2] National Language Resource Monitoring and Research Center of Minority Languages, Minzu University of China, Beijing 100081, China

[3] Key Laboratory of Ethnic Language Intelligent Analysis and Security Governance of MOE, Minzu University of China, Beijing 100081, China

[4] Language Information Security Research CenterInstitute of National Security MUC, Minzu University of China, Beijing 100081, China

Published online: 31 May 2024

provide more stable training and better mode coverage (Nichol and Dhariwal 2021; Xiao et al. 2021). Furthermore, DMs do not impose strict constraints on the model architecture as other generative models (Song et al. 2023) such as Autoregressive models (Uria et al. 2016), Variational Autoencoders (VAEs) (Kingma and Welling 2013), or Normalizing Flows (Flows) (Kingma and Dhariwal 2018). Therefore, with their huge advantages and potential, DMs have been rapidly expanded to other fields such as audio and music synthesis (Chen et al. 2020, 2022b; Okamoto et al. 2021; Liu et al. 2023a; He et al. 2023c), language model (Gong et al. 2022; Li et al. 2022a; Tang et al. 2023), video generation (Mei and Patel 2023; Ruan et al. 2023; Blattmann et al. 2023; Ni et al. 2023), etc. after being successfully applied in the field of image processing (Nichol and Dhariwal 2021; Dhariwal and Nichol 2021; Ho et al. 2020; Song et al. 2022), and achieved satisfactory results.

DMs are trained using noise-perturbed data and learn to remove corresponding noise from noisy data. The corresponding two processes of adding noise and denoising are placed on the Markov chain. Different from the single-step generation of other generative models such as GANs, VAEs, and Flows, trained DMs can obtain the results of the target data distribution in the multi-step iterative denoising process. They obtain high-quality samples in this way of gradually optimizing intermediate results coupled with larger-scale neural network architectures. But this also comes at a high price in terms of time consumption, computational cost, and storage resources. For example, GENIE used a cluster of NVIDIA V100 GPUs with a total computation of about 163k GPU hours (Dockhorn et al. 2022); PBE takes about 7 days to train using 64 NVIDIA V100 GPUs (Yang et al. 2023b); DiT-XL/2 (Peebles and Xie 2022) requires about 950 and 1733 V100 GPU days of training for 256×256 and 512×512 images, respectively (Xie et al. 2023); ResGrad and GradTTS (Popov et al. 2021) both use 8 NVIDIA V100 GPUs and require 500k and 1700k training steps, respectively (Chen et al. 2022b); DALL-E 2 contains 4 independent DMs and requires 5.5B parameters (Rombach et al. 2022; Ramesh et al. 2022); ADM has a parameter size of 552.8M on a 256×256 image synthesis, and it takes 5 days to generate 50k samples with an A100 GPU (Dhariwal and Nichol 2021; Yang et al. 2023c). The iterative generative process of DMs is typically 10 to 2000 times more computationally intensive than other single-step generative models (Song et al. 2023; Zhang and Chen 2022; Lu et al. 2022a). The improvement and application of these early DMs mainly focused on exchanging cost for high sample quality, resulting in a large number of model parameters, long research cycles, and high hardware requirements. Therefore, this situation limits the generalization of DMs in real-time demanding and resource-constrained tasks. And the high resource consumption also puts pressure on most researchers, which further hinders the exploration and development of DMs. Under this background, lightweight DMs have become a very urgent and valuable research problem (Song et al. 2023).

Recent studies have begun to pay attention to and try to solve the problem of high cost of DMs. With the input of researchers in different fields, the lightweight research of DMs has gradually achieved a satisfactory balance between processing efficiency and result quality (Luhman and Luhman 2021; Lemercier et al. 2023; Qian et al. 2022; Li et al. 2022d; Yu et al. 2023b; Mao et al. 2023; Zhang et al. 2023d; Shang et al. 2023a). However, the number of related works is gradually increasing, and the methods are also quite different. It is becoming more and more difficult for researchers to keep up with the speed of new progress, which is not conducive to the promotion and popularization of DMs. Therefore, it is urgent to investigate the progress of existing lightweight DMs. This paper will review the existing lightweight DMs methods in the field of image processing, and classify the papers according to the lightweight ideas. In addition, based on the analysis of the current research results, the future prospects of lightweight DMs methods are also given. It is hoped that this work can provide valuable reference for scholars studying DMs.

The rest of this paper is organized as follows. Section 2 discusses the fundamentals of DMs. Section 3 is the main part of this paper. Specifically, the current methods of lightweight DMs in the field of image processing are classified into eight categories: knowledge distillation (KD), quantization, pruning, fine-tuning, signal domain transformation, algorithm optimization, hybrid strategy, and other methods. This paper takes different methods of lightweighting as the main line, and selects representative work for discussion. In the final Sect. 4, a brief analysis is provided, and future prospects are discussed.

2 Basic principles of diffusion models

DMs are first applied to image generation tasks in the field of image processing. Its original idea is to sample high-quality images close to the original sample distribution from random Gaussian noise (Ho et al. 2020). It consists of a forward process (or diffusion process) responsible for gradually adding noise to the original data distribution, and a reverse process (or denoising process) that is opposite to the forward process, that is, to recover the target distribution from the noise by iterative denoising (Nichol and Dhariwal 2021). The two key processes of DMs (Sects. 2.1 and 2.2) will provide a brief description below in order to understand the methods of lightweight DMs later.

2.1 Forward process

Given a sample $\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0)$ that conforms to the distribution of the original data. The forward process is governed by a Markov chain (Ho et al. 2020). As t increases, larger Gaussian noise is gradually introduced into $\boldsymbol{x}_0$ by a variance schedule $\beta_t \in (0, 1)$ (Fig. 1):

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t\boldsymbol{I}), \tag{1}$$

where $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ represents a Gaussian transition at each step, i.e. adding noise to $\boldsymbol{x}_{t-1}$. When adding noise in the forward process until step T, $\boldsymbol{x}_T$ is obtained that approximates an isotropic Gaussian distribution. The number of diffusion steps T is set manually, and is set to 1000, 4000 in early DMs (Nichol and Dhariwal 2021; Ho et al. 2020). When $\boldsymbol{x}_0$ is given, the conditional distribution of the latent $\boldsymbol{x}_t$ at any t can be expressed in closed form by means of Eq. (1):

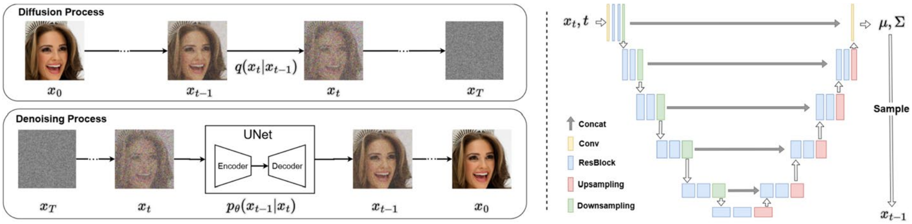


Fig. 1 Brief description of DMs. The upper and lower parts on the left of the figure represent the forward process and the reverse process, respectively. Where $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ and $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ represent a Gaussian transition in the forward and reverse processes, respectively. $\boldsymbol{x}_0$, $\boldsymbol{x}_t$ and $\boldsymbol{x}_T$ are the original image, the noisy image of the intermediate process, and pure Gaussian noise, respectively. The right of the figure is a diagram of the U-Net network responsible for prediction (Shang et al. 2023b)

$$\boldsymbol{x}_t(\boldsymbol{x}_0, \boldsymbol{\epsilon}) = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{(1 - \bar{\alpha}_t)}\boldsymbol{\epsilon}, \tag{2}$$

where $\alpha_t = 1 - \beta_t, \bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s, \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I})$. Therefore, the forward process can directly add any degree of noise to $\boldsymbol{x}_0$ in a single step without using a neural network. Combining the above Eqs. (1) and (2), when $\boldsymbol{x}_t$ and $\boldsymbol{x}_0$ of the forward process are known, the posterior distribution $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$ of $\boldsymbol{x}_{t-1}$ can be obtained by using Bayes' theorem (Nichol and Dhariwal 2021).

2.2 Reverse process

The reverse process is placed in the Markov chain opposite to that of the forward process (Fig. 1), starting at $p(\boldsymbol{x}_T) \sim \mathcal{N}(\boldsymbol{x}_T;0, \boldsymbol{I})$. To make $p_\theta(\boldsymbol{x}_0)$ fit the true distribution $q(\boldsymbol{x}_0)$, the result is refined by T-step iterations:

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_{t-1};\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t), \boldsymbol{\Sigma}_\theta(\boldsymbol{x}_t, t)), \tag{3}$$

where $\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t)$ and $\boldsymbol{\Sigma}_\theta(\boldsymbol{x}_t, t)$ represent the predicteds of the mean and variance of the distribution $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$, respectively (Nichol and Dhariwal 2021; Li et al. 2022d). The parameterization of the mean is usually modified, and the network can directly predict $\boldsymbol{x}_0$ (Xiao et al. 2021; Wang et al. 2022c; Xu et al. 2023) or $\boldsymbol{\epsilon}$ (Ho et al. 2020; Rombach et al. 2022; Yang et al. 2023c). $\boldsymbol{\Sigma}_\theta(\boldsymbol{x}_t, t)$ is generally set as a time-related constant (Ho et al. 2020). A common setting is to use the output $\boldsymbol{\epsilon}_\theta$ of network to predict $\boldsymbol{\epsilon}$, and the variance set to $\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$. In each iteration, the prediction noise $\boldsymbol{\epsilon}_\theta$ is used to denoise the intermediate results:

$$\boldsymbol{x}_{t-1} = \mathcal{N}\left(\boldsymbol{x}_{t-1}; \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\right), \tilde{\beta}_t\boldsymbol{I}\right). \tag{4}$$

The loss function is generally constructed using a variational lower bound on the negative log-likelihood. After analytical calculation and simplification of the loss, the training objective of a simple mean square error is used to train the model (Ho et al. 2020):

$$L_{simple}(\theta) = \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left[||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}, t\right)||^2\right]. \tag{5}$$

3 Lightweight methods of diffusion models

Although DMs obtain satisfactory high-quality results, the process of training and inference consumes a lot of time, computation and storage costs. Because DMs not only rely on hundreds or thousands of diffusion steps T, but also require the help of network evaluation to refine the results at each step in the sampling process (Nichol and Dhariwal 2021; Ho et al. 2020; Shang et al. 2023b). To make DMs break through the challenge of being deployed in an environment with limited storage and computing power, it is particularly important to study the lightweight of DMs. In recent years, efforts have been made on lightweight DMs. And because DMs are successfully applied in the image field for the first time, the development of lightweight DMs in this field is relatively long. Therefore, this section focuses on the methods of lightweight DMs in the field of image processing, and classifies them into KD, quantization, pruning, fine-tuning, signal domain transformation, algorithm optimization, hybrid strategy, and other methods according to the lightweight ideas used in the existing papers (Fig. 2). The following sections will describe each of these eight categories.

3.1 Knowledge distillation

Although large-scale pre-trained DMs have competitive performance, their large computation and memory consumption make practical deployment difficult. Practical devices usually have limited storage and computing capabilities and are quite different from each other. Therefore, the cost of these two aspects should be reduced as much as possible when lightweighting DMs. KD can meet this need. It is a way to compress a powerful and cumbersome sample model to another lightweight sample model without sacrificing too much

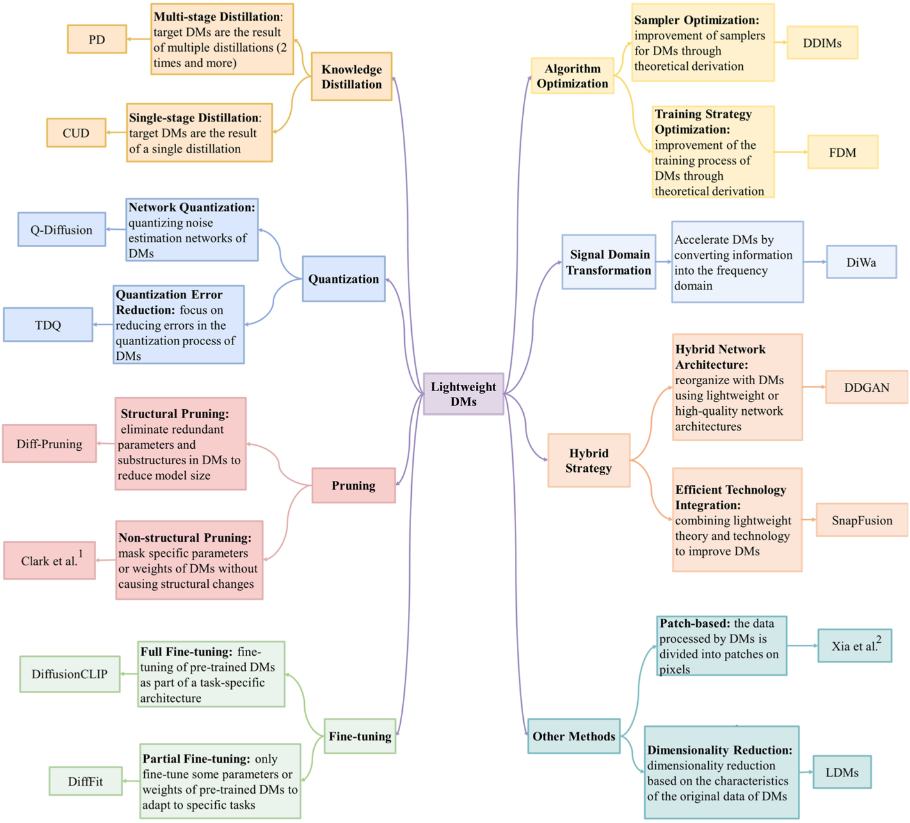

Lightweight DMs
Knowledge Distillation
Multi-stage Distillation: target DMs are the result of multiple distillations (2 times and more)
PD
Single-stage Distillation: target DMs are the result of a single distillation
CUD
Quantization
Network Quantization: quantizing noise estimation networks of DMs
Q-Diffusion
Quantization Error Reduction: focus on reducing errors in the quantization process of DMs
TDQ
Pruning
Structural Pruning: eliminate redundant parameters and substructures in DMs to reduce model size
Diff-Pruning
Non-structural Pruning: mask specific parameters or weights of DMs without causing structural changes
Clark et al.1
Fine-tuning
Full Fine-tuning: fine-tuning of pre-trained DMs as part of a task-specific architecture
DiffusionCLIP
Partial Fine-tuning: only fine-tune some parameters or weights of pre-trained DMs to adapt to specific tasks
DiffFit
Algorithm Optimization
Sampler Optimization: improvement of samplers for DMs through theoretical derivation
DDIMs
Training Strategy Optimization: improvement of the training process of DMs through theoretical derivation
FDM
Signal Domain Transformation
Accelerate DMs by converting information into the frequency domain
DiWa
Hybrid Strategy
Hybrid Network Architecture: reorganize with DMs using lightweight or high-quality network architectures
DDGAN
Efficient Technology Integration: combining lightweight theory and technology to improve DMs
SnapFusion
Other Methods
Patch-based: the data processed by DMs is divided into patches on pixels
Xia et al.2
Dimensionality Reduction: dimensionality reduction based on the characteristics of the original data of DMs
LDMs

Fig. 2 Diagram of the classification of lightweight DMs. There are eight categories, including KD, quantization, pruning, fine-tuning, signal domain transformation, algorithm optimization, hybrid strategy, and other methods. In the figure, subcategories are further divided under each main category and the characteristics of each category are explained. In addition, the corresponding representative model is attached as an example. The superscripts 1 and 2 correspond to Clark and Jaini (2023) and Xia et al. (2022), respectively

model performance (Chen et al. 2022a; Zhang et al. 2022b). The output distribution of the student model matches that of the teacher DMs for training (Song et al. 2023; Yin et al. 2022; Huang et al. 2022; Salimans and Ho 2021). Large, slow teacher DMs can then be distilled into lightweight, faster-sampling student models. Sampling can often be accelerated by reducing the number of sampling steps. Because fewer sampling steps represent fewer network calls, thereby reducing the computational overhead in use. To achieve the goal of a small number of steps, multiple or single distillations can be adopted. According to the number of distillations, these methods can be divided into multi-stage distillation and single-stage distillation.

3.1.1 Multi-stage distillation

Salimans and Ho (2021) noticed that single-stage distillation of Luhman and Luhman (2021) requires running the original model on full sampling steps to build a large dataset for training. So its distillation cost is linear with the number of sampling steps. To reduce the cost of distillation, they proposed progressive distillation (PD) to gradually reduce the number of network evaluations required for sampling (Salimans and Ho 2021). This method takes a trained sampler as a teacher model, and sets the student model to have the same architecture and number of parameters as the teacher model. A single student step is used to match two teacher steps, so that new DMs only requires half the original sampling steps. This distillation process is then continued to be applied to the student model, iteratively halving the number of required sampling steps (Fig. 3). This method can finally reduce the sampling steps to 4 steps with little performance loss (Salimans and Ho 2021). Many later works were also inspired by PD. They have improved in expanding the applicable scenarios and optimizing the distillation process.

Classifier-free guided DMs further expand the application scope of DMs and explore their potential. But such DMs need to evaluate a class-conditional and an unconditional DMs when sampling (Ramesh et al. 2022; Nichol et al. 2022; Saharia et al. 2022a), thus

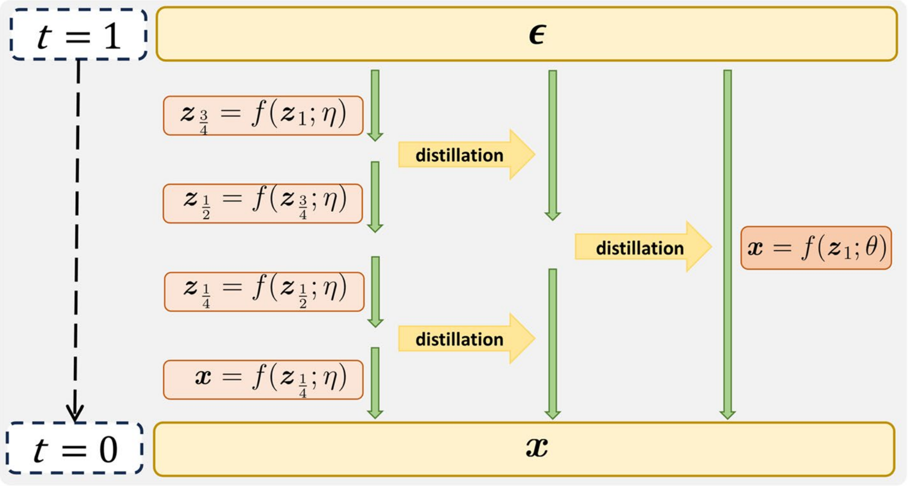

$t = 1$
ϵ
$z_{\frac{3}{4}} = f(z_1; \eta)$
$z_{\frac{1}{2}} = f(z_{\frac{3}{4}}; \eta)$
$z_{\frac{1}{4}} = f(z_{\frac{1}{2}}; \eta)$
$x = f(z_{\frac{1}{4}}; \eta)$
distillation
distillation
distillation
$x = f(z_1; \theta)$
$t = 0$
x

Fig. 3 Diagram of PD iterative reduces sampling steps. It is assumed that the original sampler $f(z;\eta)$ denoises from random noise ϵ to sample x in 4 steps. After two distillations, the new sampler $f(z;\theta)$ has 1/4 of the original steps (Salimans and Ho 2021)

requiring higher computational complexity than unconditional guidance. And lightweight processing is urgently needed. However, PD only explores the scenario of unconditional guided sampling. So Meng et al. proposed a two-stage distillation strategy to make PD applicable to classifier-free guided DMs (Meng et al. 2023). The method first learns a single model to match the combined output of the conditional and unconditional models. Then, the model is gradually distilled through the PD strategy into a model with fewer sampling steps. This work matches the performance of the teacher with only 2 to 4 sampling steps. And the effectiveness of distilling classifier-free DMs for pixel space (Ho and Salimans 2021) and latent space (Rombach et al. 2022) was also demonstrated for the first time (Meng et al. 2023).

In addition, PD needs to progressively align the output images of the T-step teacher sampler with the $T/2$-step student sampler for training. But it is difficult to directly align the images. Sun et al. proposed classifier-based feature distillation (CFD) to solve this problem (Sun et al. 2022). CFD first uses a pre-trained classifier independent of the dataset to obtain the advanced features of the output images of the student model and the teacher model. And it then computes the Kullback–Leibler (KL) divergence between the two feature distributions. This guides the student model to focus on those features that are closely related and important to image composition. And this further distills the intermediate layer outputs of the noise evaluation model (Shao et al. 2023). At the same time, CFD bypasses the problem of alignment. Therefore, the learning load of the model is reduced and the consistency of images is ensured (Sun et al. 2022). Berthelot et al. (Berthelot et al. 2023) also extended PD and proposed the transitive closure time-distillation (TRACT). Compared to PD, TRACT is not constrained by the mandatory requirement of $T' = T/2$. Each of its training stages allows distillation of T steps to arbitrary $T' < T$ steps up to the desired number of steps. In this way, TRACT reduces the number of distillation stages from $\log_2 T$ to a small constant and can achieve high-quality samples in one step (Berthelot et al. 2023).

3.1.2 Single-stage distillation

Luhman and Luhman (2021) applied KD to the lightweight of DMs for the first time. They use pre-trained DMs (Song et al. 2020a) as the teacher model, and the student model do not change the network structure. The student is trained to generate images in a single step by computing the KL divergence between the two distributions of the final output of the student and the teacher (Luhman and Luhman 2021). After that, there was a lot of work on multi-stage distillations to avoid the performance degradation of single-stage distillations. But single-stage distillation has also been getting good results recently. Song et al. proposed consistency distillation (CD) (Song et al. 2023). CD uses a numerical ordinary differential equation (ODE) solver and pre-trained DMs to generate pairs of adjacent points on a probability flow (PF) ODE trajectory (Song et al. 2023). DMs are distilled by minimizing the difference between the model outputs of these adjacent point pairs. Ultimately, DMs will be a model that maps any point at any time step to the start of a trajectory. This achieves the generation of higher quality samples than PD in fewer steps (Song et al. 2023). Shao et al. argued that previous distillation methods all required pre-trained weights, and the student architecture was heavily dependent on the chosen teacher architecture (Shao et al. 2023). It limited the scalability of the student model. Therefore, they proposed catch-up distillation (CUD) method, using DMs to simultaneously play the role of teacher and student without any pre-trained weights. CUD encourages the current moment output of the model to “catch up” with its previous moment output. At the same time, it aligns the current moment output with both the ground truth (GT) label and the previous moment output by adjusting the training target. The multi-step alignment distillation based on Runge–Kutta makes full use of all sampling points and provides more comprehensive information about $\frac{X_t}{dt}$ than one-step alignment distillation (Fig. 4). This prevents asynchronous updates, improving model performance. Experiments have found that CUD can achieve similar results with fewer iterations and a smaller batch size than CD (Shao et al. 2023).

3.2 Quantization

Quantization can also make DMs more satisfying in terms of storage and computation requirements during actual deployment. This approach replaces floating-point parameters (biases, activations, and weights) in neural networks with low-precision floating-point values or a small set of training values (Cheikh Tourad and Eleuldj 2022). It reduces the data storage and computational complexity of the model by changing the representation of data without changing the network architecture, and achieves fast inference (Bai et al. 2022). Quantization is generally divided into two categories: quantization-aware training (QAT) and post-training quantization (PTQ) (Wei et al. 2021; Youn et al. 2022). Among them, QAT usually uses the entire training set for quantization during the training stage. This requires more training time, memory requirements, and data consumption (Bai et al. 2022; Youn et al. 2022; Macha et al. 2023). In contrast, PTQ does not require full datasets or expensive retraining. Although PTQ causes a certain information loss, making the model performance slightly weaker than QAT (Bai et al. 2022; Wei et al. 2021; Macha et al. 2023; Oh et al. 2022), it is more attractive for DMs requiring expensive training costs.Therefore, most of the current work uses PTQ to quantify DMs. Specifically, it mainly promotes the quantization of DMs from two aspects: quantization noise evaluation network and quantization error reduction.

3.2.1 Network quantization

Traditional DMs need to use computationally intensive neural networks for iterative noise evaluation during the generation process. Therefore, reducing the number of network evaluations and the cost of a single evaluation can reduce the total computational overhead.

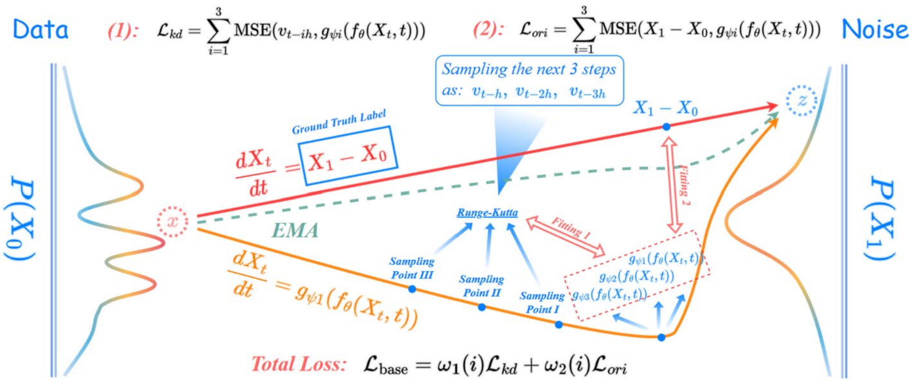
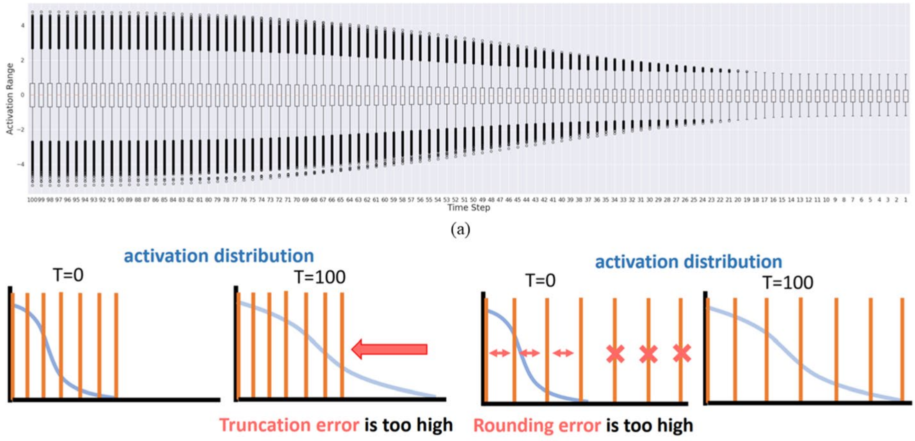

Data
(1): $\mathcal{L}_{kd} = \sum_{i=1}^{3} \mathrm{MSE}(v_{t-ih}, g_{\psi i}(f_\theta(X_t,t)))$
(2): $\mathcal{L}_{ori} = \sum_{i=1}^{3} \mathrm{MSE}(X_1 - X_0, g_{\psi i}(f_\theta(X_t,t)))$
Noise
Sampling the next 3 steps as: v_{t-h}, v_{t-2h}, v_{t-3h}
$X_1 - X_0$
Ground Truth Label
$\frac{dX_t}{dt} = X_1 - X_0$
$P(X_0)$
x
z
$P(X_1)$
EMA
Runge-Kutta
Fitting 1
Fitting 2
$\frac{dX_t}{dt} = g_{\psi 1}(f_\theta(X_t,t))$
Sampling Point III
Sampling Point II
Sampling Point I
$g_{\psi 1}(f_\theta(X_t,t))$
$g_{\psi 2}(f_\theta(X_t,t))$
$g_{\psi 3}(f_\theta(X_t,t))$
Total Loss: $\mathcal{L}_{\text{base}} = \omega_1(i)\mathcal{L}_{kd} + \omega_2(i)\mathcal{L}_{ori}$

Fig. 4 The framework of the Runge–Kutta-based multi-step alignment distillation of the CUD (Shao et al. 2023)

Both Shang et al. (2023b) and Li et al. (2023a) introduced PTQ to reduce the single cost of the evaluation by compressing the noise evaluation network. But the output distribution of the evaluation network of DMs will change with the time step. So the traditional PTQ for single time step scenario fails in DMs (Fig. 5). To solve this problem, Shang et al. explored the core design of PTQ on DMs from three aspects of quantization operation, calibration dataset and calibration metric, and proposed PTQ4DM (Shang et al. 2023b). First, the computation-intensive convolutional and fully-connected layers in the network can be quantized and batch normalization folded into the convolutional layers (Shang et al. 2023b). Operations of special functions such as SiLU and softmax maintain full-precision. Experiments show that the output $\boldsymbol{x}_{t-1}$, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ (Fig. 1) of the DMs network are not sensitive to quantization. Thus the operations that produce them can also be quantized. Second, PTQ4DM proposes normally distributed time-step calibration (NDTC) to obtain a good calibration dataset. This method samples a set of time steps from a skew normal distribution. Calibration samples are then generated using the denoising process of full-precision DMs based on this time step. In this way, the time step difference of the calibration set is enhanced to improve the model performance. Third, the mean-square error (MSE) is chosen as a measure to quantify DMs according to the experimental effect. Experimental results show that the method can maintain or even improve performance after the full-precision DMs are quantized into 8-bit models after tailoring these three aspects. Moreover, it does not require retraining and can be used in conjunction with other fast sampling methods (Song et al. 2020a). The PTQ method proposed by Li et al. to deal with the multi-time step structure of DMs is called Q-Diffusion (Li et al. 2023a). The inputs of adjacent consecutive time steps have relatively similar distributions. A small calibration set can thus be generated by randomly sampling some intermediate inputs uniformly in a fixed interval across all time steps. This balances the size of the calibration set and its representational ability distributed over all time steps. When calibrating a quantized model, the model is



(a)

(b) The interval is calibrated at T=0

(c) The interval is calibrated at T=100

Fig. 5 The range of activations of the floating-point 32 (FP32) output of the DMs (Song et al. 2020a) at different time steps (top) (Li et al. 2023a) and the limitations of static quantization of DMs (bottom) (So et al. 2023). The y axis and x axis in **a** represent the activation value and the time step in the diffusion process, respectively. It shows that the activation distribution changes gradually with the time step. Assume that the activation distribution grows gradually as the time step progresses. **b** represents a small quantization interval and a large truncation error. And **c** represents a large quantization interval and a large rounding error

divided into several reconstruction blocks. The model iteratively reconstructs the output and adjusts the clipping range and scaling factors of the weight quantizers in each block. The mean squared error between quantization and full-precision output is minimized by adaptive rounding. The core component of residual connections in the U-Net network of DMs is defined as a block. And other parts of the model that do not meet this condition are calibrated in a layer-by-layer manner. This better resolves inter-layer dependencies and generalization. For activation quantization, only the step size of the quantizer is adjusted since the activations are constantly changing with the input. Experimental results show that the method is capable of directly quantizing full-precision DMs to 8-bit or 4-bit models without losing much perceptual quality.

3.2.2 Quantization error reduction

Wang et al. also noticed that the activation distribution at different time steps and the validity variation of calibration images obtained at different time steps increase the quantization error (Wang et al. 2023b). To reduce the increase in quantization error and the decrease in model performance caused by these two problems, they proposed an efficient data-free PTQ framework for DMs (ADP-DM). To avoid the overfitting of step-wise quantization caused by limited calibration samples, ADP-DM adopts a differentiable search strategy. In this way, the optimal group assignment at different generation time steps is obtained, and the discretization function of each group is learned by minimizing the discretization error. At the same time, ADP-DM also selects the optimal time step for calibration image generation through the principle of structural risk minimization. This is different from PTQ4DM (Shang et al. 2023b) by manually specifying the time step index. This method reduces the discretization error and improves the generalization ability of quantized DMs in deployment with negligible computational overhead. Experimental results show that the proposed method achieves a significant boost in performance compared to PTQ methods for DMs (Shang et al. 2023b; Li et al. 2023a) with similar computational cost (Wang et al. 2023b). However, So et al. argued that although previous studies (Shang et al. 2023b; Li et al. 2023a) focused on the dynamics of the input activation distribution, they still relied on static parameters (So et al. 2023). This leads to suboptimal convergence in minimizing the quantization error. Therefore, they proposed a new method for activation quantification in DMs, temporal dynamic quantization (TDQ). The TDQ module generates a time-dependent optimal quantization configuration, dynamically adjusts the quantization interval according to the time step information. This minimizes activation quantization errors, significantly improving output quality. Unlike conventional dynamic quantization techniques, this method has no computational overhead during inference and is compatible with both PTQ and QAT (So et al. 2023).

At the same time as ADP-DM, He et al. aimed to systematically analyze the impact of quantification on DMs, and also established a unified PTQ framework (He et al. 2023a). They argued that the quantization noise in each denoising step leads to a bias in the estimated mean. And quantization noise may also be accumulated during the sampling process, damaging the quality of the generated samples. Therefore, they proposed a unified formula for the quantization noise and diffuse perturbed noise. That is, the quantization noise of the original quantized noise prediction network is decomposed into $(1 + k)\epsilon_\theta(\boldsymbol{x}_t, t) + \Delta_{\epsilon_\theta(\boldsymbol{x}_t,t)}$. Where $k\epsilon_\theta(\boldsymbol{x}_t, t)$ is the linear correlation part corresponding to full precision and $\Delta_{\epsilon_\theta(\boldsymbol{x}_t,t)}$ is the residual uncorrelated part. These two parts are corrected separately by estimating the correlation coefficient and correcting the denoising variance schedule to reduce the mean bias and extra variance in each denoising step. In addition, they also proposed a mixed-precision scheme to select the optimal bitwidth of each denoising step. Specifically, the speedup and high signal-to-noise ratio (SNR) are guaranteed by early low bits and late high bits in the denoising steps. Experiments show that the proposed method can generate higher quality samples than Q-Diffusion (Li et al. 2023a) at a lower computational cost.

3.3 Pruning

There are non-contributing parameters or structures in the DMs network structure, which is also one of the reasons that affect the size of the model and the speed of inference. Pruning can eliminate non-key parameters in the model and ignore some computational costs with low returns (Chin et al. 2020). It can be divided into structural pruning and non-structural pruning. Structural pruning effectively reduces the model size by eliminating redundant parameters and substructures in the network, while non-structural pruning essentially masks parameter by zeroing them out (Fang et al. 2023a; Sanh et al. 2020). By pruning DMs, a balance between accuracy and speed can be achieved, minimizing accuracy reduction and maximizing speed. This enables models to be deployed and run more efficiently in resource-constrained environments.

3.3.1 Structural pruning

Considering the iterative nature of DMs when generating samples, Fang et al. adaptively improved the structural pruning technology according to their characteristics, and proposed Diff-Pruning (Fang et al. 2023b). Different diffusion time steps contribute differently to the content and detail of the sample (Rombach et al. 2022; Yang et al. 2023c). And noise levels with large t cannot provide information gradients. So Diff-Pruning models the trade-off between content, detail, and noise as a pruning problem of diffusion time steps. The method utilizes Taylor expansion to identify and prune non-contributing diffusion steps to provide an efficient and stable approximation of partial steps. Furthermore, the full Taylor expansion does not accurately estimate the weight importance due to the accumulation of some noisy gradients from the convergence gradients or unimportant steps. A thresholding strategy with binary weights α_t is therefore introduced to determine the weights, modeling the pruning problem as a weighted trade-off between content and detail (Fig. 6). The

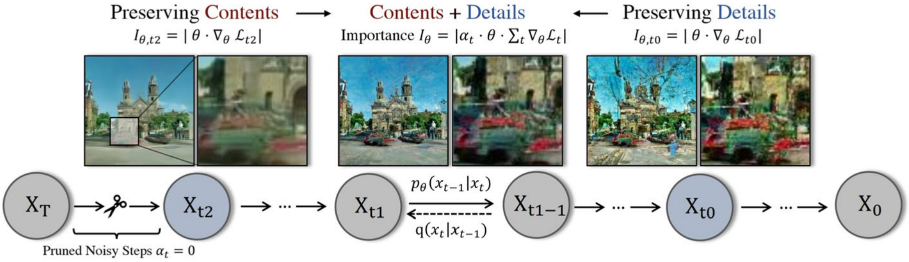

Preserving Contents
$I_{\theta,t2} = |\,\theta \cdot \nabla_\theta\, \mathcal{L}_{t2}|$
Contents + Details
Importance $I_\theta = |\alpha_t \cdot \theta \cdot \sum_t \nabla_\theta \mathcal{L}_t|$
Preserving Details
$I_{\theta,t0} = |\,\theta \cdot \nabla_\theta\, \mathcal{L}_{t0}|$
X_T
X_{t2}
X_{t1}
$p_\theta(x_{t-1}|x_t)$
$q(x_t|x_{t-1})$
X_{t1-1}
X_{t0}
X_0
Pruned Noisy Steps $\alpha_t = 0$

Fig. 6 The pruning strategy of Diff-Pruning. The Taylor expansion is exploited to identify and remove non-critical steps at time steps that require pruning. Where binary weights $\alpha_t \in \{0, 1\}$ weigh local details (such as edges and colors), and contents (such as objects and shapes) (Fang et al. 2023b). (Color figure online)

lightweight process of the model only needs 10% to 20% of the training cost, which is much smaller than the training cost of the original model (Fang et al. 2023b).

3.3.2 Non-structural pruning

In early work, Li et al. pruned model weights by zeroing out elements smaller than a self-set threshold to achieve the target sparsity (Li et al. 2022b). This method directly uses the pre-trained weights and uniformly prunes 40% of the model weights. This fine-grained pruning method does not outperform the effect of its hybrid strategy due to no further fitness improvements.

Transferring pre-trained DMs to downstream tasks can not only fully exploit the potential of the original model, but also save the training cost for new tasks. At the same time, in the process of transferring tasks, pruning can also be used to further reduce the cost, making the new model easier to use. Clark et al. proposed a method for evaluating DMs for Text-to-Image (TTI) as zero-shot classifiers (Clark and Jaini 2023). This method uses the ability of DMs to denoise noisy images as a proxy for the likelihood of the label given the text description of the label, thereby achieving the evaluation of the zero-shot classification task. But this classification process requires multiple denoising of images for each class. With the help of pruning, this method cuts out the obviously wrong categories in advance, greatly reducing computing resources. This method effectively applies DMs for TTI to other tasks besides generation, revealing more capabilities of pre-trained DMs (Clark and Jaini 2023).

3.4 Fine-tuning

Models with higher performance or adapted to new tasks can be obtained through specialized architectural design and training from scratch. However, in addition to cost constraints, this may also be limited by insufficient training data or the difficulty of generating complex scenes with high quality (Wang et al. 2022b). Large pre-trained DMs have competitive performance. Transferring them to new tasks can circumvent these problems to a certain extent and save resource consumption during training. Fine-tuning can initialize a new model with the weights of a pre-trained model and further adjust and train on a new

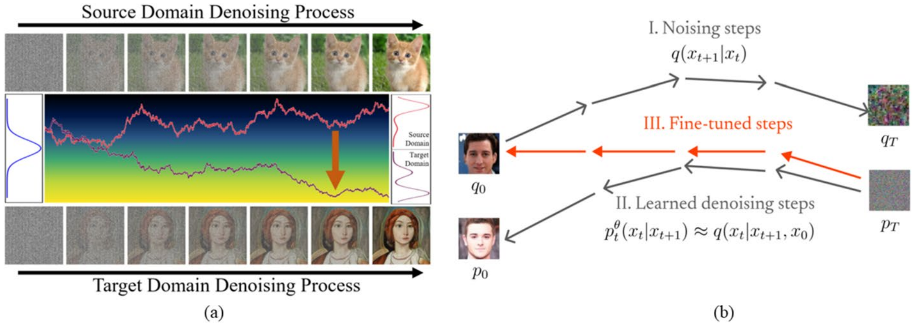

Source Domain Denoising Process
Source Domain
Target Domain
Target Domain Denoising Process
(a)
I. Noising steps
$q(x_{t+1}|x_t)$
III. Fine-tuned steps
q_T
q_0
II. Learned denoising steps
$p_t^\theta(x_t|x_{t+1}) \approx q(x_t|x_{t+1}, x_0)$
p_T
p_0
(b)

Fig. 7 Diagram of source and target domains in DMs. **a** shows that the denoising process of the DMs usually generates images iteratively from random noise, so that large DMs pre-trained in the source domain can be fine-tuned to the target domain (Xie et al. 2023). **b** shows that improved data can be generated by exploring other unexplored paths without following the reverse process (Fan and Lee 2023)

dataset (Mahajan et al. 2018; Wortsman et al. 2022; Liu et al. 2022; Church et al. 2021). Therefore, fine-tuning DMs as another lightweight direction allows users to reuse large pre-trained DMs while minimizing computation and resource usage (Fig. 7). According to the degree of adjustment, fine-tuning can be divided into full fine-tuning and partial fine-tuning. Where full fine-tuning usually refers to fine-tuning all parameters of the pre-trained model, while partial fine-tuning is the opposite. Appropriate selection based on specific task requirements can cost-effectively improve the performance of DMs and expand their application breadth (Kumar et al. 2022; Kumari et al. 2023).

3.4.1 Full fine-tuning

The process of transferring the knowledge learned by pre-trained DMs to downstream tasks will be limited by dealing with personalized topics and the risk of overfitting (Han et al. 2023). Kim et al. proposed DiffusionCLIP, which can perform image editing guided by text prompts, and successfully performs zero-shot image processing even between unseen domains (Kim et al. 2022b). The method utilizes pre-trained DMs and the loss of the contrastive language-image pretraining (CLIP) model for text-guided image processing. They found that directly fine-tuning the ϵ_θ that the DMs is responsible for predicting is more effective than fine-tuning the latents. Therefore, a loss consisting of the directional CLIP loss $L_{direction}$ and the consistency loss L_{id} is set. As shown in Fig. 8, the $L_{direction}$ loss is calculated by the original image $\boldsymbol{x}_0$, the predicted image $\hat{\boldsymbol{x}}_0$ generated by the latent, the reference text $\boldsymbol{y}_{ref}$, and the target text $\boldsymbol{y}_{tar}$ to supervise the optimization. And L_{id} is used to ensure the characteristics of the object to avoid unnecessary changes. Thus, the generation quality of fine-tuned DMs is guaranteed. Wang et al. paid more attention to comprehensive pre-training models that are easier to transfer to downstream tasks, and thus proposed a general framework based on pre-trained DMs (Wang et al. 2022b). DMs are first pre-trained conditioned on semantic inputs. A large number of different types of images helps DMs have the ability to act as generative

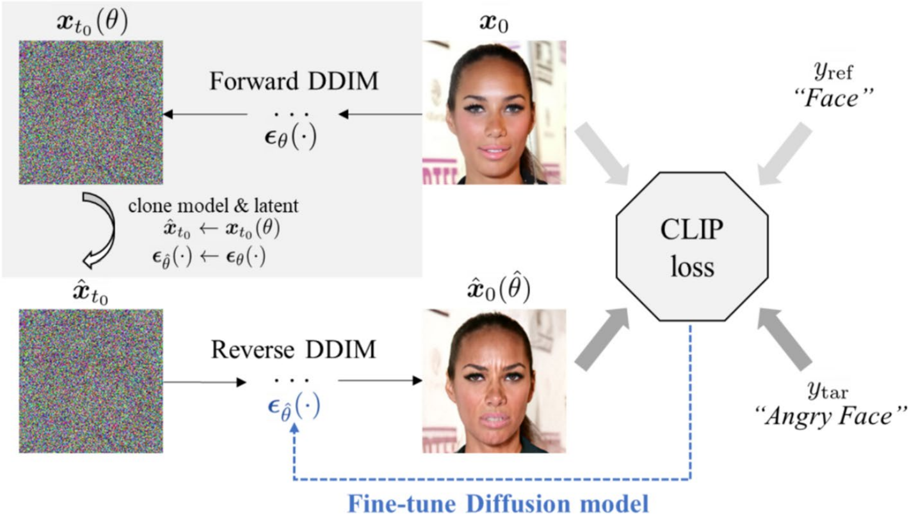

$x_{t_0}(\theta)$
x_0
Forward DDIM
$\epsilon_\theta(\cdot)$
y_{ref}
"Face"
clone model & latent
$\hat{x}_{t_0} \leftarrow x_{t_0}(\theta)$
$\epsilon_{\hat{\theta}}(\cdot) \leftarrow \epsilon_\theta(\cdot)$
CLIP loss
$\hat{x}_{t_0}$
$\hat{x}_0(\hat{\theta})$
Reverse DDIM
$\epsilon_{\hat{\theta}}(\cdot)$
y_{tar}
"Angry Face"
Fine-tune Diffusion model

Fig. 8 Diagram of DiffusionCLIP. The input images are first transformed into latents by DMs. Then, the fine-tuned DMs predict the updated samples guided by the directional CLIP loss (Kim et al. 2022b)

priors for general images. And this does not require any task-specific customization or hyperparameter tuning. The trained DMs can be used as decoders. A two-stage fine-tuning scheme is followed to maximize the use of pre-trained knowledge and adapt to downstream tasks. In the first stage the decoder is fixed. Only a task-specific encoder needs to be trained to map conditional inputs to a pre-trained latent space. The entire model is jointly fine-tuned in the second stage to improve the overall performance.

Large DMs for TTI lack the ability to imitate content in a given reference set and synthesize new features in different contexts (Ruiz et al. 2023a). Therefore, Ruiz et al. proposed a method for personalization generation, DreamBooth, to make better use of pre-trained DMs (Ruiz et al. 2023a). Its goal is to introduce a new “unique identifier, subject” pair into the “dictionary” of the model. This enables category-specific prior knowledge to be combined with the subject in order to generate new images of the subject. Directly fine-tuning all layers of the model may reduce output diversity. And it will also bring about the problem of language drift, that is, the model will gradually forget how to generate topics of the same category as the target topic. Therefore, DreamBooth generates samples using samplers on frozen pre-trained DMs. A self-generated class-specific prior preservation loss is added to supervise the model output (Fig. 9). Experiments show that the fine-tuning method only needs 3–5 subject images to be able to synthesize subjects under different conditions that are not present in the reference images. This makes large pre-trained DMs easier to personalize. The diffusion policy optimization with KL regularization (DPOK) further optimizes the image quality of DMs for TTI (Fan et al. 2023). This method defines the fine-tuning task as a reinforcement learning (RL) problem. And it avoids computing gradients with trajectories that may cause storage inefficiencies, but instead uses policy gradients to update pre-trained DMs to maximize the reward for feedback training. To prevent overfitting the reward model, the KL divergence between the fine-tuned model and the pre-trained model is used as a regularizer. Experiments show that DPOK generally outperforms simple supervised fine-tuning methods (Lee et al. 2023c) in image-text alignment and image quality. Later, DomainStudio (Zhu et al. 2023) was proposed based on DreamBooth (Ruiz et al. 2023a). In the fine-tuning process, the method uses the high-frequency components in the source domain and the target domain to perform pairwise similarity loss

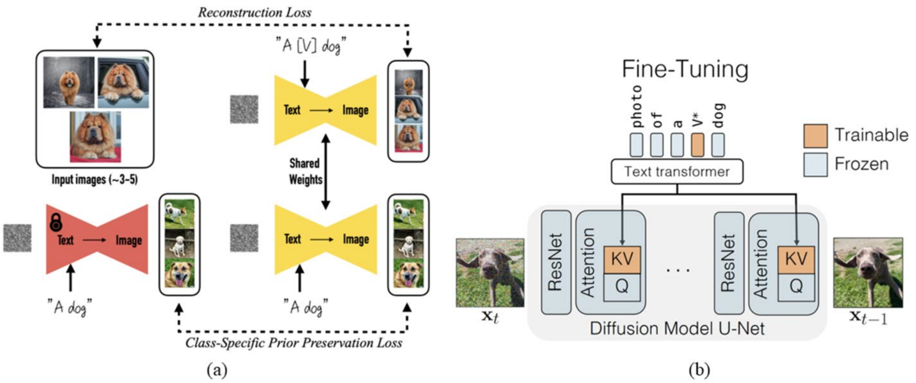


Fig. 9 Diagram of DreamBooth and Custom Diffusion. DreamBooth fine-tunes all layers of the model (**a**, Ruiz et al. 2023a), and Custom Diffusion only fine-tunes the key and value parameters of the cross-attention layers (**b** Kumari et al. 2023)

to enhance the diversity of high-frequency details in the samples. And it uses high-frequency reconstruction loss to enhance the learning of high-frequency details in samples, thereby improving the generation quality.

Shortcut fine-tuning (SFT) focuses more on the sampling speed after fine-tuning (Fan and Lee 2023). The method models trajectories of conditional probabilities. It provides an alternative method of gradient estimation equivalent to policy gradient, which does not require differentiation through composite functions. Specifically, the pre-trained DMs sampler is fine-tuned by directly minimizing the integral probability metric (IPM), instead of learning the backward diffusion process. This allows the sampler to find a more efficient sampling shortcut than the backward diffusion process without changing the noise distribution (Fig. 7) (Fan and Lee 2023).

Jia et al. argued that previous methods (Kumari et al. 2023; Ruiz et al. 2023a) were slow to process each object and that the storage cost of the model increased with the number of objects to process (Jia et al. 2023). To reduce storage requirements and speed up fine-tuning without compromising the performance of large pre-trained DMs, they focus on bypassing lengthy optimizations. An encoder is first adopted to capture the high-level identifiable semantics of objects, and only a single feed-forward pass is required to generate object-specific embeddings. Then, the obtained object embeddings are passed to the TTI synthesis model for subsequent generation (Jia et al. 2023). This model aims to generalize to unknown objects. Therefore, it is not suitable for fine-tuning a small number of parameters in the way of Custom Diffusion (Kumari et al. 2023), and needs to fine-tune the enhanced network as a whole.

3.4.2 Partial fine-tuning

Custom Diffusion has a similar goal to DreamBooth (Kumari et al. 2023). But instead of fine-tuning all parameters, it fine-tunes a small number of parameters. The optimization backbone of this method is stable diffusion (SDMs) (Rombach et al. 2022). Although the cross-attention layer parameters only account for 5% of the overall parameters, this part has a high impact on the latent features of the model. Therefore, only the parameters of the cross-attention layer need to be fine-tuned. Whereas the fine-tuning task aims to update the mapping from given text to image distribution, and the text features are only fed into the key and value parameter matrices in the cross-attention block. Therefore, this method only needs to update a small subset of parameters consisting of the keys and values of the cross-attention layers (Fig. 9). To prevent language drift, single concept fine-tuning of this method does not modify the loss like DreamBooth. Instead, it is solved using a regularized dataset containing images whose titles have a high similarity to the target text prompt. Furthermore, merging the key and value matrices of multiple text features for training can allow multiple fine-tuning concepts to be generated. This method greatly reduces the time of the fine-tuning, enabling fast tuning of models to represent new concepts in about 6 min, efficiently enhancing existing models. Also based on SDMs, Wu et al. proposed a simple and lightweight image editing algorithm to achieve style matching and content preservation of images (Wu et al. 2023a). The whole process optimizes the hybrid weights of two text embeddings under two objectives, one is a perceptual loss for content preservation and the other is a CLIP-based style matching loss. The method only involves optimizing about 50 parameters without fine-tuning the DMs themselves. Experiments show that the method can modify a large range of attributes without affecting other content, and outperforms DMs-based image editing algorithms that require fine-tuning (Kim et al. 2022b).

SVDiff is different from the above methods in the idea of reducing the risk of overfitting and language drift (Han et al. 2023). Inspired by FSGAN, this method fine-tunes the singular values of the weight matrix of pre-trained DMs to obtain a compact and efficient spectral shift parameter space. Specifically, instead of fine-tuning the entire weight matrix, fine-tuning is performed on the spectral shift, that is, the difference between the singular values of the updated weight matrix and the original weight matrix. Separately trained spectral shifts can be combined with each other to form new models for the generation of different image styles. After experimental comparison, the model size of SVDiff is 1/2 to 1/3 of that of LoRA (Hu et al. 2021), another fine-tuning method.

Xie et al. proposed a parameter-efficient fine-tuning strategy, DiffFit, to adapt large pre-trained DMs to various downstream domains (Xie et al. 2023). This method only fine-tunes the bias term, normalization, class-conditional modules, and scaling factors that accommodate feature scaling enhancements. Compared to full fine-tuning, DiffFit achieves a 2× training speed-up and only needs to store about 0.12% of the total model parameters.

3.5 Signal domain transformation

The high-dimensional characteristics and redundancy of image information increase the processing volume of DMs. The signal domain transformation can convert images processed by DMs from the spatial domain to the frequency domain. In this way, targeted operations can be performed on different frequency components, thereby reducing the burden of model training and sampling. Techniques such as the Fourier transform (FT) or wavelet transform for frequency domain processing are backed by mathematical principles. And they have also been verified to be effective in image processing tasks (Zhao et al. 2023c; Shen et al. 2023). This provides a new perspective for lightweight DMs.

Guth et al. believed that the high-dimensional probability distribution of natural images has complex multi-scale characteristics, and the time consumed increases rapidly with the increase of image size (Guth et al. 2022). Thus, they proposed a wavelet score-based generative model (WSGM). This method can decompose the image into the product of conditional probabilities of normalized wavelet coefficients across scales to simplify the computation. WSGM first generates the first low-resolution (LR) image. And this generated LR image is conditionally renormalized by the wavelet coefficients. Then a higher resolution image is reconstructed from these wavelet coefficients by fast inverse wavelet transform (IWT). This process repeats the reverse diffusion on the discrete wavelet coefficients at each scale to obtain higher and higher resolution images. And the number of steps on each scale is the same. This can be orders of magnitude smaller than original DMs (Ho et al. 2020).

Further, Phung et al. found that although DDGAN (Xiao et al. 2021) can greatly shorten the running time of the model by reducing the sampling steps from thousands to several steps, its speed is still far behind that of GANs (Phung et al. 2023). To further reduce the speed gap while maintaining the generation of high-quality images, they used wavelet decomposition to extract low-frequency and high-frequency components from the image and feature layers. These components are processed adaptively to speed up processing without loss of generation quality (Fig. 10). Meanwhile, a reconstruction term is introduced to improve the convergence of model training (Phung et al. 2023). Zhang et al. found that when the coordinates of the target dataset have widely different scales or are strongly correlated, the sampling process will be ill-conditioned (Zhang et al. 2023c). Therefore, although a larger step size can reduce the number of steps, the

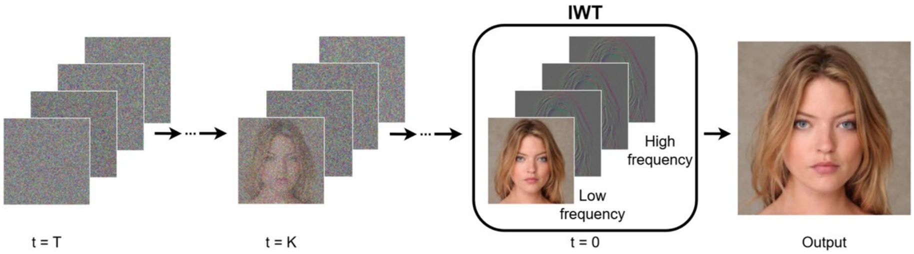


Fig. 10 Wavelet-based diffusion scheme for (Phung et al. 2023). The denoising operation is performed in the wavelet space. And after T steps of denoising, the final result in pixel space is reconstructed through IWT (Phung et al. 2023)

convergence of the sampling process cannot be guaranteed. They proposed a model-agnostic preconditioned diffusion sampling (PDS) method. The sampling process is introduced into a preconditioning matrix to make the scale of the target distribution more similar across all coordinates. And fast FT is used to reduce construction cost. This method balances the scales of different coordinates in the sample space, alleviating the ill-conditioned problem without retraining the model.

Moser et al. proposed a diffusion-wavelet (DiWa) method (Moser et al. 2023) to reduce memory consumption and processing time. This method combines the advantages of DMs and discrete wavelet transformation (DWT), and is applied to image super-resolution reconstruction (SR) task. There are two benefits to DMs working in the DWT domain. The first is that DWT directly isolates high-frequency details in separate subbands. This makes the representation more sparse, which is good for model learning. The second is that DWT halves the spatial size of the image according to the Nyquist rule (Moser et al. 2023). This speeds up the per-step inference time of the denoising network and improves the convergence rate. Experiments show that the parameters of this method are 17% and 78% of those of SR3 (Saharia et al. 2022b) and SRDiff (Li et al. 2022c) previously applied to SR tasks, respectively. The diffusion-based low-light (DiffLL) method (Jiang et al. 2023b) is used for low-light image enhancement tasks and also introduces DWT. It contains wavelet-based conditional DMs (WCDM) to solve the problems of large consumption of computing resources and unstable restoration effect of traditional DMs in image restoration (IR) tasks. The low-light image is converted to the wavelet domain by 2D-DWT twice to obtain the average coefficient and high-frequency coefficient. WCDM performs a diffusion operation on the average coefficient for robust and efficient recovery. The high-frequency restoration modules (HFRM) supplement the diagonal details with vertical and horizontal information to obtain high-frequency coefficients. This is used to coordinate reconstruction of fine-grained details (Jiang et al. 2023b). Finally, the inverse 2D-DWT is used to transform the results into pixel space. WCDM speeds up the inference process while maintaining perceptual fidelity, reducing the usage of computing resources (Jiang et al. 2023b).

3.6 Algorithm optimization

The above methods benefit from lightweighting techniques that have proven effective on other models. They were adaptively modified and applied to DMs with remarkable results.

There are currently many works on research based on the characteristics of DMs. They obtain effective training strategies and efficient samplers by optimizing the algorithms of the training and sampling process, thereby improving model performance and efficiency.

3.6.1 Sampler optimization

Reducing the number of iteration steps in the sampling process can lightweight DMs. However, this can also increase the step size, resulting in a serious drop in performance. To solve this problem and achieve high-quality samples with fewer sampling steps, a lot of work on optimization algorithms has been generated. Song et al. proposed denoising diffusion implicit models (DDIMs) (Song et al. 2020a). This method generalizes the Markovian forward process used by the original DMs (Ho et al. 2020) to a non-Markovian forward process. This allows DMs to uniformly skip some steps in the sampling process to reduce the number of sampling steps. Finally, a shorter reverse Markov chain is obtained (Fig. 11). And DDIMs do not need to change the objective function. Therefore, pre-trained DMs can be used directly, avoiding retraining the model. Experiments show that the method generates high-quality samples at 10× to 50× the sampling speed of the original DMs (Ho et al. 2020). Ghimire et al. explained this problem from a geometric perspective (Ghimire et al. 2023). They proved that the addition of noise in the space of probability measures and the forward and reverse processes of DMs are Wasserstein gradient flows. However, when the number of sampling steps is reduced, the samples will generate errors at each step and gradually move away from the gradient flow path. This results in a higher overall error. Thus, they proposed an estimation of Wasserstein Gradient. And the intermediate samples are projected back to the gradient flow path after each step to guarantee the descent towards the gradient flow path. ACDMSR adds a deterministic denoising process to improve performance and speed up the inference process (Niu et al. 2023). A pre-trained SR model is used to provide a pre-super-resolved version $\hat{\boldsymbol{x}}_0$ of a given LR image. Similar to (Preechakul et al. 2022), the obtained $\hat{\boldsymbol{x}}_0$ can replace $\boldsymbol{x}_0$ in the deterministic iterative denoising process $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$ to achieve SR and assist in generating images with more visual fidelity (Saharia et al. 2022b; Li et al. 2022c).

Meng et al. still use skip steps as the main idea to reduce the number of sampling steps, and propose stochastic differential editing (SDEdit) for image synthesis and editing (Meng et al. 2021). Unlike the uniform skipping of DDIMs (Song et al. 2020a), this method does not start denoising from Gaussian noise, but performs noise perturbation on the scribbled image input (Fig. 12). This perturbation result serves as a prior for DMs and is gradually synthesized into realistic images. This method regulates the sampling time by the degree of prior perturbation, and synthesizes realistic images with fewer steps. Similarly, Zhang et al. also skipped the previous steps to reduce the sampling steps, and proposed a retrieval-based diffusion sampling framework (ReDi) (Zhang et al. 2023e). But ReDi retrieves trajectories

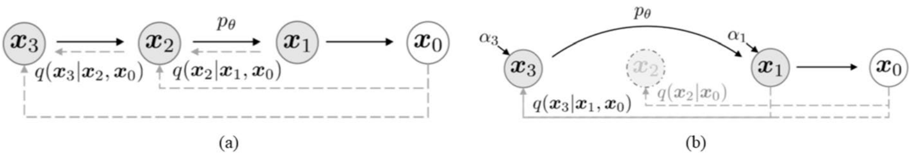


Fig. 11 Diagram of DDIMs. **a** shows a non-Markovian diffusion process. **b** shows that the sampling process that originally required three steps can be completed in two steps with the help of DDIMs (Song et al. 2020a)

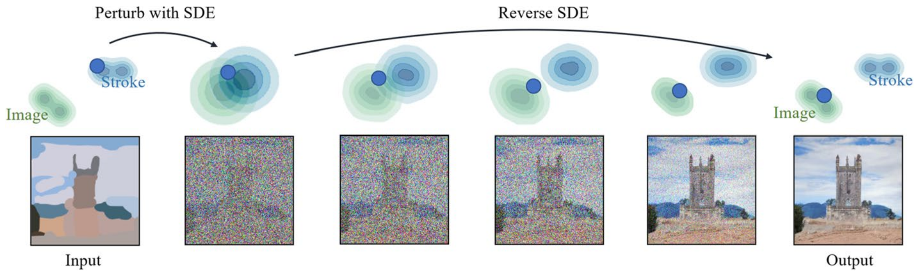


Fig. 12 Diagram of SDEdit. Blue dots represent edits to the image. The green and blue contour lines represent the distribution of images and strokes, respectively (Meng et al. 2021). (Color figure online)

similar to partly generated trajectories from a pre-computed knowledge base. This part is then skipped in the early stages of the generation, speeding up the inference process. Afterwards, the model continues to sample from later steps of the retrieved trajectories. Unlike SDEdit (Meng et al. 2021), CoreDiff does not add noise to the input image (Gao et al. 2023a). This method replaces Gaussian noise with an operator that simulates physical degradation. Therefore, it can directly use the information-rich image to replace the random Gaussian noise at the beginning of the sampling process, skipping the previous sampling step. ResShift is a fast sampling model designed for SR (Yue et al. 2023). It also does not use white Gaussian noise, but starts with a prior distribution based on LR images. A new Markov chain is designed, which makes the residuals of HR images and LR images gradually transferred. Combined with noise schedule, it is possible to more precisely control changes in residuals and noise levels during transitions. This method avoids the problem of over-blurry SR results produced by previous accelerated sampling techniques (Rombach et al. 2022; Song et al. 2020a), and outperforms them in efficiency. ResShift requires only 15 sampling steps to achieve better results than previous methods (Zhang et al. 2021; Liang et al. 2022).

Gao et al. considered that the inference process of the original DMs (Ho et al. 2020) was approximated by solving the corresponding diffusion ODE (Song et al. 2020b) in the continuum limit (Gao et al. 2023b). Therefore, they investigated a numerical analysis method for ODE solvers, namely backward error analysis. Moreover, a fast sampling scheme based on dynamically adjusting the long-term backward error is proposed, which is called the restricting backward error (RBE) table. They found that backward error analysis was useful for clarifying the role of finite steps and helping to identify implicit biases of different ODE solvers (Gao et al. 2023b). Its goal is to describe the bias introduced when integrating an ODE with finite steps by introducing a modified ancillary flow. This ensures that the discrete iterations of the original ODE are on the path of the continuous solution of the modified flow. According to the RBE table, the model is able to generate samples that outperform early samplers (Ho et al. 2020; Lu et al. 2022a; Song et al. 2020a) within 8 sampling steps without any training. Permenter et al. argued that learning denoising is related to learning projections (Permenter and Yuan 2023). Therefore, the sampling of DMs is reinterpreted as an approximate gradient descent applied to the Euclidean distance function. And the convergence analysis of the sampler of DDIM is provided under the assumption of the projection error of the denoiser. Using the properties of the distance function, a high-order sampler is designed. This sampler aggregates previous denoiser outputs to reduce errors. This method enables the sampler to obtain high-quality samples in 5 to 10 steps on pre-trained DMs (Ho et al. 2020; Song et al. 2020a).

Lu et al. (2022b) and Wizadwongsa and Suwajanakorn (2022) mainly focused on methods for reducing the number of steps for guided sampling. They both found that the first-order solver (Song et al. 2020a) can perform guided sampling to improve sample quality, but requires a larger number of sampling steps. Higher-order samplers can sample in fewer steps without guidance. But it performs worse than low-order methods when applied to guided sampling (Wizadwongsa and Suwajanakorn 2022). To apply high-order samplers to guided sampling and obtain higher-quality samples with a small number of steps, they explored from different aspects. Lu et al. found that large guidance scales shrink the convergence radius of high-order solvers, making them unstable (Lu et al. 2022b). And its converged solution is not in the same range as the original data. Therefore, high-order samplers will have problems of instability and slow speed. The high-order solver (DPM-Solver++) proposed by them is not to predict the noise $\boldsymbol{\epsilon}_\theta$, but to use the data prediction model $\boldsymbol{x}_\theta$ from $\boldsymbol{x}_t$ to predict $\boldsymbol{x}_0$ to solve the diffusion ODE. And the solution is matched with the training data distribution by a threshold method, so as to maintain the quality of sampling. Wizadwongsa et al. found that the difference between guided sampling and unguided sampling in classical high-order methods is whether the gradient of the conditional function is added to the sampling equation (Wizadwongsa and Suwajanakorn 2022). Therefore, they conjectured that classical higher-order methods may not be suitable for conditional functions. A solution based on the operator splitting method of strang splitting (Strang 1968) is proposed. It separates the underperforming condition function term from the standard diffusion term and solves it separately at each time step. This method allows use in various conditional generation tasks, such as TTI, image inpainting, colorization and SR. Zhao et al. wanted to further accelerate the high-order solver and designed a sampling framework using a pre-trained model called UniPC (Zhao et al. 2023a). UniPC consists of a corrector (UniC) and a predictor (UniP) in the same analytical form. Inspired by the prediction-correction method for solving ODE values, UniPC can be applied after previous DMs samplers. And it does not require additional model evaluation to improve sampling accuracy. The framework has a uniform analytical form for arbitrary orders and supports unconditional and conditional sampling.

Unlike the above methods, Golnari et al. did not reduce the cost by reducing the sampling steps (Golnari et al. 2023). They focused on optimizing SDMs guided inference pipeline and proposed a method to simplify the noise computation process during sampling. Initial iterations establish the general layout of the image, and later iterations work on improving its overall quality. Therefore, limiting optimization to late iterations avoids any adverse impact on the overall performance of the model. The predicted noise contains unconditional and conditional noise terms. Computational complexity can be reduced by removing unconditional noise for some iterations. Experiments show that optimization extended to 50% iterations can reduce inference time by 20.3%.

3.6.2 Training strategy optimization

Li et al. started from the problem and performed selective calculations (Li et al. 2022b). They argued that only local regions of pixels need to be updated for image editing at any moment. But DMs resynthesize regions that don't need to be modified, which wastes computational resources. Therefore, a general method called spatially sparse inference (SSI) is proposed. To selectively compute on edited regions, first all activations of the original input image need to be precomputed. During editing, the edited region is located by the calculated difference mask between the original image and the edited image (Li et al. 2022b). Unedited regions can then reuse precomputed activations, while edited regions are updated simply by applying convolutional filters. Experiments show that the method can reduce the computation of DDIMs (Song et al. 2020a) by 7.5 times and speed up the sampling time by 6.6 times while maintaining the visual fidelity when the editing area accounts for 1.2% of the total area.

To speed up the convergence speed of DMs training and shorten the training cycle, Hang et al. proposed a weighting strategy called Min-SNR-γ for optimization (Hang et al. 2023). They found that the optimization direction conflicts between the time steps of DMs, which caused the optimization process to be blocked. To improve the convergence speed of the model, they considered diffusion training as a multi-task learning problem. That is, the denoising process of each step is regarded as a separate task. And the loss weight of each task is determined by its task difficulty, that is, the compressed SNR. This effectively balances conflicts between different time steps. Experimental results show that the proposed method converges 3.4 times faster than previous weighting strategies (Gu et al. 2022). At the same time, better image generation quality can be achieved while using a smaller architecture than previous state-of-the-art (Song et al. 2020a; Bao et al. 2023). Voronov et al. pointed out that the previous textual inversion (TI) method (Ruiz et al. 2023a) had a long training time, which limited the feasibility of practical applications and increased the experimental time of research (Voronov et al. 2023). They found that most concepts are already learned in the early stages, and the quality does not improve in the later stages. To speed up the training process, deterministic variance evaluation (DVAR) is proposed, which is an early stopping criterion of TI. DVAR uses batches of entirely random data to train the model but evaluates it on batches with partially fixed randomness across all iterations. This method improves the speed of adaptation by 15 times without significant degradation in image quality (Voronov et al. 2023). Fast DMs (FDM) improve the forward process from a stochastic optimization perspective to speed up training and sampling (Wu et al. 2023b). The authors noticed that the forward process conforms to the stochastic optimization process of stochastic gradient descent (SGD) for a stochastic time-variant problem (Wu et al. 2023b). Meanwhile, momentum SGD can achieve more stable and faster convergence. Therefore, FDM introduces a momentum mechanism to improve the forward process. Moreover, the momentum SGD process is regarded as a damped oscillation system, and the noise perturbation kernel function is derived. This avoids oscillation and achieves a faster convergence rate of the forward process. Experimental results show that FDM can be applied to DM frameworks such as VP (Song et al. 2020b), VE (Song et al. 2020b) and EDM (Karras et al. 2022).And FDM reduces their training cost by about half and the sampling steps by a factor of about 3. DMs model score evolution with a single time-varying neural network, resulting in long training time and limited modeling flexibility (Haxholli and Lorenzi 2023). Haxholli et al. proposed a parallel score matching strategy to address this issue (Haxholli and Lorenzi 2023). The method employs separate neural networks to separately learn the evolution of scores within a specific time sub-interval. It is further extended to use different networks to independently model the scores of each individual time point. Experimental results show that this approach significantly speeds up the training process through data parallelization and additional parallelization layers. ProtoDiffusion combines prototype learning and DMs (Baykal et al. 2023). Specifically, it first learns prototypes of the classes using a separate classifier. It then introduces these prototypes into the training of DMs as conditional information to guide the diffusion process. Experiments show that learned prototypes can have an impact on model performance in the early stages of training, thereby speeding up training.

Ning et al. pointed out that as the step size increases, sampling will accumulate errors and reduce the accuracy of the model (Ning et al. 2023). These errors are mainly due to the difference between the training phase and the sampling phase. During training, $\boldsymbol{x}_t$ is given to predict $\boldsymbol{x}_{t-1}$. However, the sampling is based on the prediction of the previous generation results, so the real $\boldsymbol{x}_t$ cannot be obtained. To alleviate this problem, they propose a DMs training regularization method (DDPM-IP) to explicitly model prediction errors. DDPM-IP does not require changes to the network structure or specific loss functions. The error of the prediction network is simulated using a dedicated random noise vector, and $\boldsymbol{x}_t$ containing error noise is provided to the prediction network. Experiments show that the model with reduced error allow larger step size, helping to reduce training and inference time. Lee et al. argued that the slow sampling efficiency of DMs is due to the high curvature of the forward process directly related to the truncation error of the numerical solver (Lee et al. 2023b). Zero curvature means that generative ODEs can be solved exactly with only one function evaluation. They found that it was the intersection between forward trajectories that caused the high curvature, and that reducing the degree of intersection improved the curvature. Therefore, they parameterized the coupling as a neural network. And the KL term is used to ensure the effective coupling between the noise distribution and the original data distribution (Fig. 13).Decoupled DMs (DDM) are designed to improve the solution efficiency of the reverse process by simplifying the forward process (Huang et al. 2023a). DDM decomposes the complex diffusion process into two relatively simple processes. The image distribution is approximated by an explicit transition probability, and the noise path is controlled by a standard Wiener process. This enables the model to learn to predict noise and image components, respectively. Furthermore, the explicit transition probability is introduced to model the gradient of image component. This allows the sampling step size to be increased. Experiments have proven that DDM has a lower computational budget for sampling than previous DMs and can generate high-quality images in 10 steps (Ho et al. 2020; Song et al. 2020b; Vahdat et al. 2021; Dockhorn et al. 2021). PartDiff approximates the intermediate state of the denoising process of the HR image to the latent state of the LR image, and starts denoising from the intermediate distribution (Zhao et al. 2023b). And the proposed latent alignment mechanism is used to gradually interpolate the latent states of LR and HR images during the training process. This compensates for the approximation error introduced by skipping a large number of denoising steps.

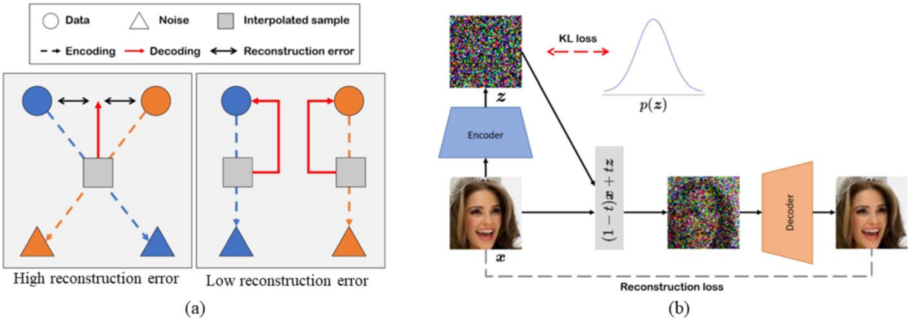


Fig. 13 The basis and vision principle of Lee et al. (2023b). **a** shows that the reconstruction error is high when the forward trajectories intersect and is low when they do not intersect. **b** shows the visual principle of the method (Lee et al. 2023b)

3.7 Hybrid strategy

Hybrid strategy is a method of integrating other architectures and technologies into DMs. It aims to take advantage of various methods to speed up the sampling of DMs or reduce the training cost when applied to other tasks. By exploring the hybrid strategy, different technologies can be more fully utilized. This can improve the effectiveness and applicability of DMs, and further promote the development of its related applications.

3.7.1 Hybrid network architecture

Combined with lightweight or high-quality model network architecture, the shortcomings of DMs can be compensated. This enables DMs to obtain high-quality results with low computational cost and expands its application scenarios. Kulikov et al. designed a lightweight fully convolutional denoiser to simplify the DMs architecture, and proposed SinDDM for single image generation (Kulikov et al. 2023). The fully convolutional denoiser of this model is constrained by both noise level and scale. The receptive field of the denoiser is set relatively small to control it to only capture statistics of fine details within each scale. Therefore, samples of any size can be generated from coarse to fine. In addition to learning the internal statistical information of the image through a multi-scale diffusion process, SinDDM can also be guided by external supervision. It generates high-quality samples while having good generalization ability. Zheng et al. proposed DMs sampling with neural operator (DSNO) from the perspective of parallel decoding (Zheng et al. 2023b). This method constructs the neural operator backbone of DSNO by introducing temporal convolutional layers parameterized in Fourier space in given DMs. And the initial Gaussian distribution is mapped to the continuous-time solution trajectory of the reverse diffusion process. This allows the final solution to be generated from a single model evaluation (Fig. 14). To extend pre-trained DMs to specific tasks and save training cost, a neural network structure named ControlNet is proposed (Zhang and Agrawala 2023). This structure is used to control pre-trained large-scale DMs to support additional input conditions. ControlNet clones the weights of large DMs into trainable and locked copies. The two are used to learn conditional control on task-specific datasets and preserve the ability of the original network learning, respectively. The trainable and locked modules are connected with zero convolutions, where the convolution weights are gradually grown from zero to optimized parameters in a learned manner. Experiments show that the model can be trained on personal devices and can maintain good robustness (Zhang and Agrawala 2023). Shang et al. noticed that simple convolutional neural networks (CNN) can save costs, and proposed a ResDiff based on the residual structure (Shang et al. 2023a). They used CNNs instead of DMs to restore the main low-frequency content, and used DMs to predict the residual between real images and CNN-predicted images. And a loss function based on frequency domain is introduced to promote the recovery ability of CNN. Frequency-domain guided diffusion enables DMs to predict high-frequency details. To improve the customization efficiency of DMs for TTI (Kumari et al. 2023; Ruiz et al. 2023a), Xiao et al. investigated the low-rankness of the multi-head attention (MHA) layer in the model (Xiao et al. 2023a). They found that the partial weight matrices of MHA did not exhibit sufficient low-rank properties. This limits the overall potential performance gain from low-rank decomposition. Therefore, the low-rankness at the head level is designed, reducing the parameters of MHA and relaxing the low-rank constraint. This method has faster training speed

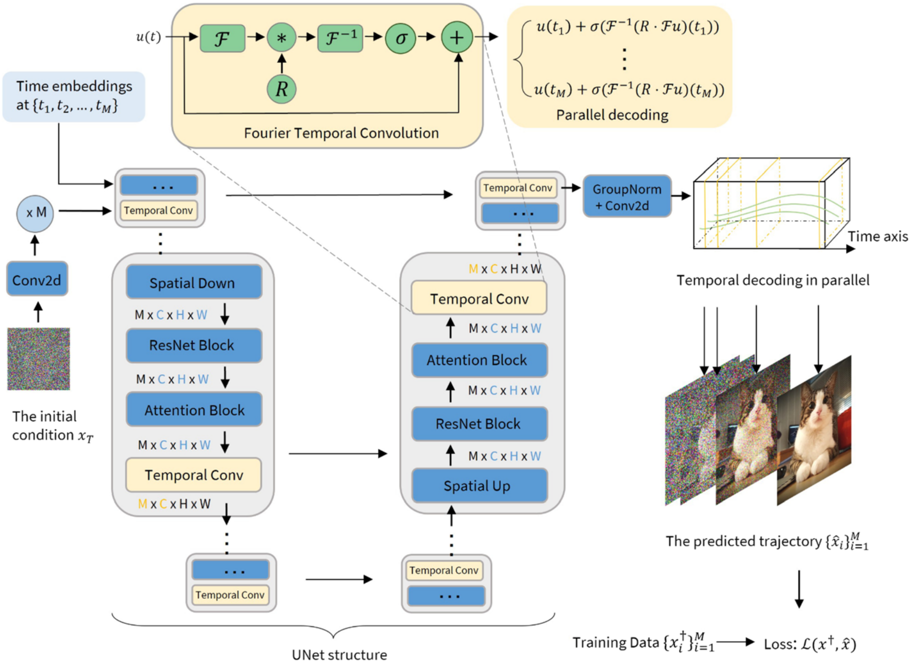

$u(t)$
$\mathcal{F}$
$*$
$\mathcal{F}^{-1}$
σ
$+$
R
Fourier Temporal Convolution
$u(t_1) + \sigma(\mathcal{F}^{-1}(R \cdot \mathcal{F}u)(t_1))$
$u(t_M) + \sigma(\mathcal{F}^{-1}(R \cdot \mathcal{F}u)(t_M))$
Parallel decoding
Time embeddings at $\{t_1, t_2, \ldots, t_M\}$
x M
Conv2d
The initial condition x_T
Temporal Conv
Spatial Down
M x C x H x W
ResNet Block
Attention Block
Temporal Conv
Spatial Up
GroupNorm + Conv2d
Time axis
Temporal decoding in parallel
The predicted trajectory $\{\hat{x}_i\}_{i=1}^M$
Training Data $\{x_i^\dagger\}_{i=1}^M$
Loss: $\mathcal{L}(x^\dagger, \hat{x})$
UNet structure

Fig. 14 The network design and training for DSNO. DSNO operates on temporal and channel dimensions with the help of the proposed temporal convolutional layer. And it outputs representations of different temporal positions in the trajectory on a single forward pass. Other blocks operate on pixel and channel dimensions (Zheng et al. 2023b)

and lower additional storage cost than previous methods (Kumari et al. 2023; Ruiz et al. 2023a).

Bond-Taylor et al. considered that DMs generate higher resolution images with higher computational requirements (Bond-Taylor et al. 2022). Therefore, they designed DMs for vector-quantized image representation to alleviate this problem. This method uses an unconstrained transformer architecture as the backbone of the DMs. This enables parallel prediction of vector quantization tokens, facilitating the unconditional generation of globally consistent high resolution (HR) and diverse images with lower computational overhead. This method is 88 times faster than the original DMs (Ho et al. 2020) after combining the operation of uniform skip steps. Hoogeboom et al. considered that methods focusing on latent diffusion in low-dimensional spaces or image generation in multi-level cascades (Rombach et al. 2022; Ho et al. 2022; Balaji et al. 2022) added additional complexity to the diffusion framework (Hoogeboom et al. 2023). Applying DMs in the pixel space of HR images remains challenging. Therefore, a single-stage model for TTI is proposed to generate HR images while maintaining the simplicity of the model (Hoogeboom et al. 2023). Low computational intensity leads to low accelerator utilization, while large activations cause out-of-memory problems. Therefore, LR feature maps are scaled to increase utilization and relieve memory pressure. And the convolutional layer with the self-attention module in the original U-Net architecture is replaced by a multi-layer perceptron module to speed up training. Zheng et al. utilized masked training to significantly reduce the training cost of DMs without sacrificing generative performance (Zheng et al.

2023a). Specifically, a large proportion (e.g., 50%) of patches in diffused input images are randomly masked during training. Masked training employs an asymmetric encoder-decoder architecture consisting of a transformer encoder that operates only on unmasked patches and a lightweight transformer decoder for full patches. And the task of reconstructing masked patches is augmented to learn the score of unmasked patches. This facilitates long-range understanding of full patches. Experiments on the ImageNet-256 × 256 dataset show that the method achieves the same level of performance as the state-of-the-art diffusion transformer (DiT) model (Peebles and Xie 2022) using 31% of the original training time (Zheng et al. 2023a). Multi-architecture multi-expert (MEME) framework adopts multiple architectures to adapt to specific frequency requirements between different time steps to improve computational efficiency (Lee et al. 2023a). Some previous works have introduced multi-expert strategies to assign denoisers to different noise intervals (Balaji et al. 2022; Go et al. 2023). But they ignore specialized operations for specific frequencies. For example, self-attention operations and convolutions are good at processing low-frequency and high-frequency components, respectively. Therefore, MEME tailors multiple experts of specialized architectures for the required operations at each time-step interval. Experiments show that MEME outperforms previous baseline models (Rombach et al. 2022) in terms of both computational efficiency and generation performance.

Autoencoders (AE) can perform representation learning on input information and are also used to lightweight DMs. For example, Preechakul et al. believed that since DMs can only approximate $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ through Gaussian distribution, a large number of sampling steps are required (Preechakul et al. 2022). When the information of $\boldsymbol{x}_0$ in $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$ is captured in large quantities, $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$ can be modeled to obtain high-quality samples faster. A method combining AE and DMs is proposed, called Diffusion AE (Diff-AE) (Preechakul et al. 2022). Diff-AE first relies on AE to extract a meaningful and decodable

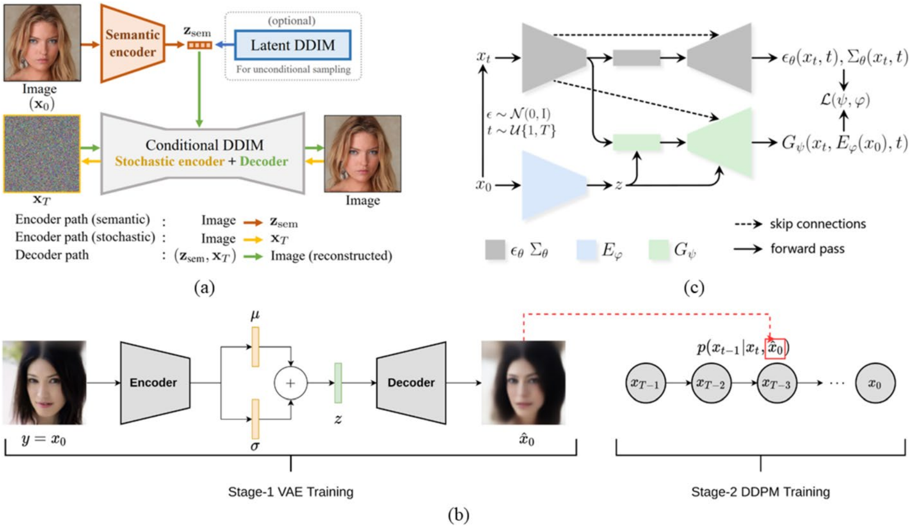

Semantic encoder
$\mathbf{z}_{sem}$
(optional)
Latent DDIM
For unconditional sampling
Image ($\mathbf{x}_0$)
Conditional DDIM
Stochastic encoder + Decoder
$\mathbf{x}_T$
Image
Encoder path (semantic) : Image → $\mathbf{z}_{sem}$
Encoder path (stochastic) : Image → $\mathbf{x}_T$
Decoder path : $(\mathbf{z}_{sem}, \mathbf{x}_T)$ → Image (reconstructed)
(a)
x_t
$\epsilon_\theta(x_t, t), \Sigma_\theta(x_t, t)$
$\mathcal{L}(\psi, \varphi)$
$\epsilon \sim \mathcal{N}(0, \mathrm{I})$
$t \sim \mathcal{U}\{1, T\}$
$G_\psi(x_t, E_\varphi(x_0), t)$
x_0
z
skip connections
forward pass
ϵ_θ Σ_θ
E_φ
G_ψ
(c)
μ
Encoder
Decoder
z
σ
$y = x_0$
$\hat{x}_0$
$p(x_{t-1}|x_t, \hat{x}_0)$
x_{T-1}
x_{T-2}
x_{T-3}
x_0
Stage-1 VAE Training
Stage-2 DDPM Training
(b)

Fig. 15 Diagram of Diff-AE (Preechakul et al. 2022), DiffuseVAE (Pandey et al. 2022) and PDAE (Zhang et al. 2022a). **a** shows that the AE in Diff-AE maps the input image $\boldsymbol{x}_0$ to the semantic encoding z_{sem} as a condition for DMs to generate images (Preechakul et al. 2022). **b** is a diagram of the two-stage training of DiffuseVAE (Pandey et al. 2022). **c** shows the training of PDAE. Pre-trained DMs are frozen during training, indicated by gray sections (Zhang et al. 2022a)

representation z_{sem} of the input image to discover high-level semantics. The conditioned DMs are then used as decoders to model the rest of the random variation (Fig. 15). Since z_{sem} captures a lot of information about $\boldsymbol{x}_0$, $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, z_{sem})$ is closer to $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$ than $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$. DMs lack low-dimensional and interpretable latent spaces, which can be compensated by VAEs (Pandey et al. 2022). Therefore, DiffuseVAE is designed as a two-stage conditional framework, using VAEs to accelerate the sampling of DMs (Pandey et al. 2022). In the first stage, the original image $\boldsymbol{x}_0$ is modeled as $\hat{\boldsymbol{x}}_0$ by standard VAEs. In the second stage, $\boldsymbol{x}_0$ is reconstructed with the DMs model conditional on $\hat{\boldsymbol{x}}_0$ (Fig. 15). Zhang et al. aimed to utilize pre-trained DMs for image reconstruction (Zhang et al. 2022a). Therefore, a general method of pre-trained DMs autoencoding (PDAE) is proposed based on Diff-AE (Zhang et al. 2022a). The information loss in the forward process of pre-trained DMs leads to a gap between the posterior mean predicted by DMs and the true mean. Therefore, the idea of PDAE is to perform representation learning through AE, that is, $E_\psi(\boldsymbol{x}_0)$ learns compact and meaningful representations in input images to help fill in the gaps of information loss (Fig. 15). The pre-trained unconditioned DMs are then used for image reconstruction. Experiments show that AE improves the training efficiency and performance of DMs. The training time required by PDAE to complete representation learning is less than half of that of Diff-AE, but the effect is still better than that of Diff-AE (Zhang et al. 2022a).

The network of models such as Flows is reversible, so model evaluation and inversion calculations are fast, and can be combined with DMs with stronger expressive capabilities (Kingma and Dhariwal 2018). Diffusion Flows (DiffFlow) connects the two (Zhang and Chen 2021). The algorithm consists of two neural stochastic differential equations (SDEs). A forward SDE gradually adds noise to the data, transforming the data into Gaussian random noise. A backward SDE gradually removes noise for sampling. The two neural SDEs are jointly trained by minimizing the KL divergence. Backward SDE starts from a Gaussian distribution and eventually converges to the desired data distribution. Compared with DMs, DiffFlow learns a more flexible forward diffusion, which can adaptively transform data into noise more effectively (Fig. 16). Subsequently, the data-adaptive implicit nonlinear DMs (INDM) also combines Flows and DMs (Kim et al. 2022a). It extends linear DMs to nonlinear diffusion processes. INDM learns nonlinear diffusion in data space

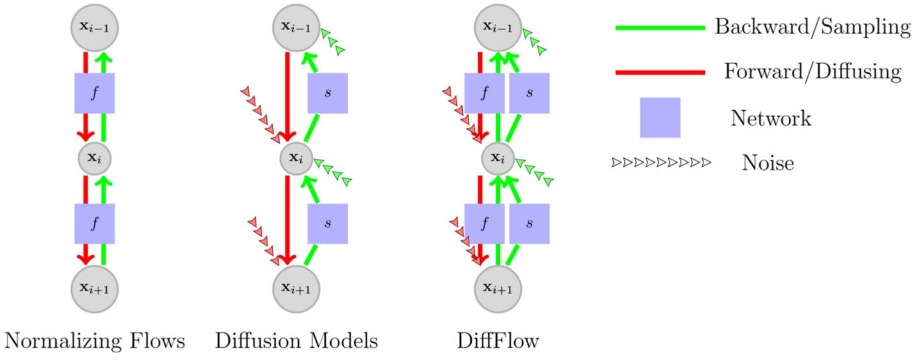


Fig. 16 Diagram of Flows, DMs and DiffFlow (Zhang and Chen 2021). Both the forward and backward processes of Flows are deterministic, while both processes of DMs are stochastic. And the forward and backward processes of DiffFlow are both trainable and stochastic (Zhang and Chen 2021)

and realizes linear diffusion in latent space through Flows. This flexible nonlinear diffusion has a better learning curve and thus is faster to train than prior work (Song et al. 2020b).

GANs are used by lightweight DMs for their fast generation speed and powerful expressive ability. Xiao et al. believed that the denoising process of DMs was limited by the Gaussian distribution and could not adapt to the denoising of large step size (Xiao et al. 2021). Therefore, they proposed to model the Gaussian distribution instead of complex multimodal distribution. Conditional GANs can simulate such complex conditional distributions in images, which GANs just lack the characteristics of mode coverage and sample diversity possessed by DMs. Denoising diffusion GANs (DDGAN) combines GANs and DMs (Xiao et al. 2021). To build adversarial training, the generator only needs to predict $\boldsymbol{x}_0$ and use $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)$ to get $\boldsymbol{x}_{t-1}$. The discriminator takes $\boldsymbol{x}_{t-1}$, $\boldsymbol{x}_t$ as input and decides whether $\boldsymbol{x}_{t-1}$ is the real version (Fig. 17). Diffusion-GAN also combines GANs (Wang et al. 2022c). It not only avoids the expensive sampling chain of DMs, but also generates more realistic images than GANs. The discriminator of this method also learns to distinguish diffused real data from generated data. But the generator learns from the feedback of discriminator by backpropagating along the forward diffusion chain, producing samples that can fool the discriminator at any diffusion step. Li et al. combined the optimization efficiency of GANs with the predictive power of DMs (Li et al. 2022d). They proposed a pixel spread model (PSM) for image inpainting of large missing regions. The model iteratively employs a decoupled probabilistic modeling strategy and predicts an outcome representing the mean term and an uncertainty map representing the variance term. Implicit adversarial training is used to optimize the mean term, resulting in more accurate predictions with fewer iterations. And the variance term is explicitly modeled using Gaussian regularization, making the model faster and lighter. Xu et al. noticed that DDGAN (Xiao et al. 2021) does not force the reverse step to use a parametric distribution, speeding up sampling with larger step size (Xu et al. 2023). But there are scalability issues when dealing with large-scale

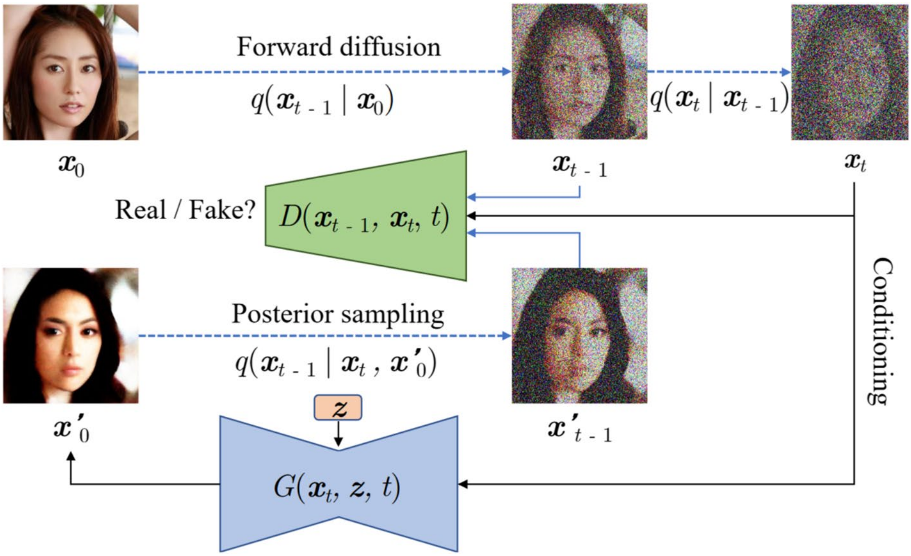
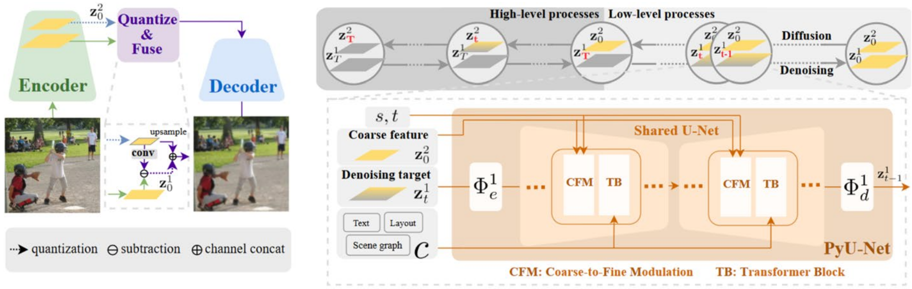

Forward diffusion
$q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_0)$
$q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$
$\boldsymbol{x}_0$
$\boldsymbol{x}_{t-1}$
$\boldsymbol{x}_t$
Real / Fake?
$D(\boldsymbol{x}_{t-1}, \boldsymbol{x}_t, t)$
Posterior sampling
$q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}'_0)$
$\boldsymbol{z}$
$\boldsymbol{x}'_0$
$\boldsymbol{x}'_{t-1}$
$G(\boldsymbol{x}_t, \boldsymbol{z}, t)$
Conditioning

Fig. 17 Diagram of training DDGAN. The green part is the discriminator, and the blue part is the generator (Xiao et al. 2021). (Color figure online)

datasets. Therefore, they further proposed an implicit model to match the marginal distribution of noisy data with the explicit conditional distribution of forward diffusion (Xu et al. 2023). This combines implicit and explicit training objectives such that the marginal distributions of random variables can be matched during the reverse diffusion process. Experiments demonstrate that the method achieves generative performance comparable to models based on DMs (Nichol and Dhariwal 2021; Song et al. 2020b; Karras et al. 2022) and outperforms models with fewer sampling steps (Xiao et al. 2021).

Frido is a kind of feature pyramid DMs (Fan et al. 2023a). It alleviates the slow inference of DMs by performing a multi-scale coarse-to-fine denoising process. Specifically, Frido uses the VQGAN-based (Esser et al. 2021) multi-scale vector quantization model (MS-VQGAN) to encode the input image into vector quantization features of multiple scales in the latent space to provide different levels of image information. Then the latents of all scales are merged, giving DMs a coarse-to-fine prior to decode the output image (Fig. 18). Ryu et al. proposed pyramid DMs (PDDPM) (Ryu and Ye 2022) to make the neural network lighter. The authors found that without positional encoding, the model lost the ability to predict the correct image at different resolutions. Therefore, PDDPM adds the condition of position information through position embedding during training, and generates HR images from coarser resolution images. The flexibility of the model is further improved by patch-based training when generating larger-scale images. Traditional DMs maintain constant resolution during sampling, which reduces speed. Therefore, Zhou et al. proposed pyramid DMs (PyDiff) for low-light image enhancement (Zhou et al. 2023b). The pyramid structure of this method gradually increases the resolution to reduce the computational cost during the sampling process and accelerate the sampling. Xia et al. found that traditional DMs can generate each pixel by starting from noise (Xia et al. 2023b). This is inefficient for IR tasks where most of the pixels are already given. Thus, they proposed DiffIR consisting of three components, including the compact IR prior extraction network (CPEN), dynamic IR transformer (DIRformer) and a denoising network for DMs. In the pre-training stage, real images are fed into CPEN to obtain a compact IR prior representation (IPR). In the second stage, training DMs directly estimates the same IPR as the pre-trained CPEN using only low-quality (LQ) images. DIRformer utilizes IPR to recover LQ images. Multiple networks are combined, which further reduces the impact of evaluation errors. And since IPR is just a compact vector, DiffIR can use fewer iterations than



(a) Architecture of MS-VQGAN.

(b) Details of the diffusion and denoising processes.

Fig. 18 Diagram of Frido (Fan et al. 2023a). **a** shows that MS-VQGAN encodes images into multi-scale features. The upper part of **b** is a process from coarse to fine, with the upper layers to lower layers denoising in sequence. The bottom half of **b** describes each denoising step. U-Net is shared at both time step and scale level (Fan et al. 2023a)

traditional DMs (Rombach et al. 2022) to obtain accurate estimates, yielding more stable and realistic results.

3.7.2 Efficient technology integration

The sampling speed of DMs can also be greatly improved by mixing with other efficient technologies. For example, Wang et al. proposed a latent feature DMs (LDDPM) similar to LDMs (Rombach et al. 2022) to solve the problem of excessive sampling time (Wang et al. 2022a). Firstly, the encoder is used to encode the image effectively, so as to reduce the solution space of the reconstructed image. Then DMs are used as generators in adversarial training, which are optimized by the discriminator to improve the generative modeling ability of the model. Bolya et al. noted that the computational scale of DMs for transformer-based backbone is related to the square of the number of tokens (Bolya and Hoffman 2023). Therefore, they proposed token merging (ToMe) to reduce the number of tokens and speed up the generation process. The generated images generally have a high redundancy. It is a waste of resources to perform calculations for each token. ToMe reduces the computational cost by combining redundant tokens to reduce tokens without any additional training. Experiments show that ToMe can ensure the performance of SDMs while increasing the generation speed by 2 times and reducing memory consumption by 5.6 times (Bolya and Hoffman 2023).

Li et al. lightened the cost of a single sampling and reduced the number of sampling steps (Li et al. 2023b). Combining network architecture and distillation, they proposed SnapFusion (Li et al. 2023b). For the research on the UNet architecture in SDMs, the authors found that the model parameters were concentrated in the downsampling stage. The slowest parts are the input and output stages with maximum feature resolution. SnapFusion uses robustly trained models and constructed evolution action set to perform online network changes of UNet, reducing redundant parts. And using the 32-step output data of the original model (Rombach et al. 2022) to perform two step distillations (Salimans and Ho 2021), an 8-step efficient UNet can be obtained. This greatly reduces the amount of computation. This method realizes running DMs for TTI in less than 2 s on mobile devices for the first time, and promotes the application process of DMs in edge devices. Kim et al. focused on reducing the cost of a single sampling and proposed a block-removed knowledge-distilled SDMs (BK-SDMs) (Kim et al. 2023). This study removes some residual blocks and attention blocks from U-Net for SDMs. This reduces the number of parameters, multiply-accumulate operations (MACs) per sampling step, and latency by more than 30%. And with limited resources, the input of U-Net can be obtained with the help of pretrained-and-frozen encoders, and then the distillation operation is performed.

3.8 Other methods

Besides the methods mentioned above, there are other works such as resizing the input image by patches, or dimensionality reduction such as signal decomposition and processing in latent spaces. They can effectively reduce the time and computational overhead of a single network evaluation, thereby further reducing the burden of the training and sampling process.

3.8.1 Patch-based

The input and output of traditional DMs are both HR images, which makes the memory requirement in the sampling process too high. A method for patch-based DMs was proposed by Xia et al (Xia et al. 2022). This method trains the DMs network by extracting patches from the original image, and then samples each patch of the image separately for image reconstruction during the sampling process. Data processing is patch-based, so workflows can be distributed in parallel, overcoming memory issues with large-scale data.

Similarly, Arakawa et al. believed that large-scale input would cause memory consumption when passing through the self-attention mechanism in the DMs network, and also proposed a DMs based on patch-wise generation (Arakawa et al. 2023). Specifically, two regulation methods are introduced. The first uses position-wise conditioning using a one-hot representation to ensure that generated patches are in the proper position. The second is global content conditioning (GCC), which ensures that patches connected together have consistent content. Experiments show that this method enables a moderate trade-off between maximum memory consumption and generated image quality. It maintains comparable image quality even when maximum memory consumption is cut in half (Arakawa et al. 2023).

3.8.2 Dimensionality reduction

Zhang et al. only focused on high dimensions in the early stages of sampling (Zhang et al. 2023b). They believed that this stage had spatial redundancy in the image signal, so it did not need to maintain high dimensionality. Thus, they proposed dimensionality-varying diffusion process (DVDP). The image is decomposed by the signal into multiple orthogonal components, and the attenuation of each component is controlled when the image is perturbed. Insignificant components can be reduced by increasing the noise intensity. The original image can be represented by a low-dimensional signal to reduce computational cost, which hardly loses information.

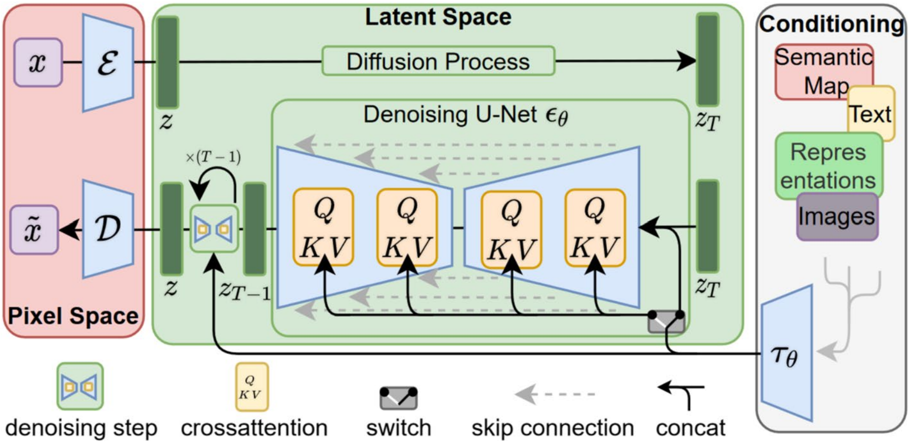


Fig. 19 Diagram of LDMs (Rombach et al. 2022). The green part is the latent space, and the red part on the left is the pixel space (Rombach et al. 2022). (Color figure online)

Latent diffusion models (LDMs) avoid the high computational cost of DMs in pixel space (Rombach et al. 2022). This method utilizes pre-trained AE to convert pixel space into latent space, reducing the complexity. The AE encoder first compresses the image to the latent space. DMs can then be diffused and sampled in the latent space. Finally, the result is restored to the original pixel space using the AE decoder (Fig. 19). This method of compressing high-dimensional features into low-dimensional spaces for operations significantly reduces computational requirements compared to pixel-based DMs (Dhariwal and Nichol 2021). Based on LDMs, Avrahami et al. run latent diffusion in a lower-dimensional latent space to speed up the inference process of the model (Avrahami et al. 2023). But each step performs local image editing by mixing latents according to user-supplied masks. This enables localized text-driven editing on generic images. Experiments show that the method is faster than previous works (Nichol et al. 2022; Avrahami et al. 2022) and produces more accurate results. Wang et al. noted that the cost and difficulty of accurately regressing pixel values increases with resolution (Wang et al. 2023a). The latent space is lower dimensional than the pixel space, which can reduce the cost of each denoising step. But regressing to real-value latent representations remains complex. Therefore, they proposed a method to represent and generate images using a binary latent space. The bidirectional mapping between an image and the corresponding latent binary representation is modeled by a trained AE with a Bernoulli encoded distribution. This binary latent space

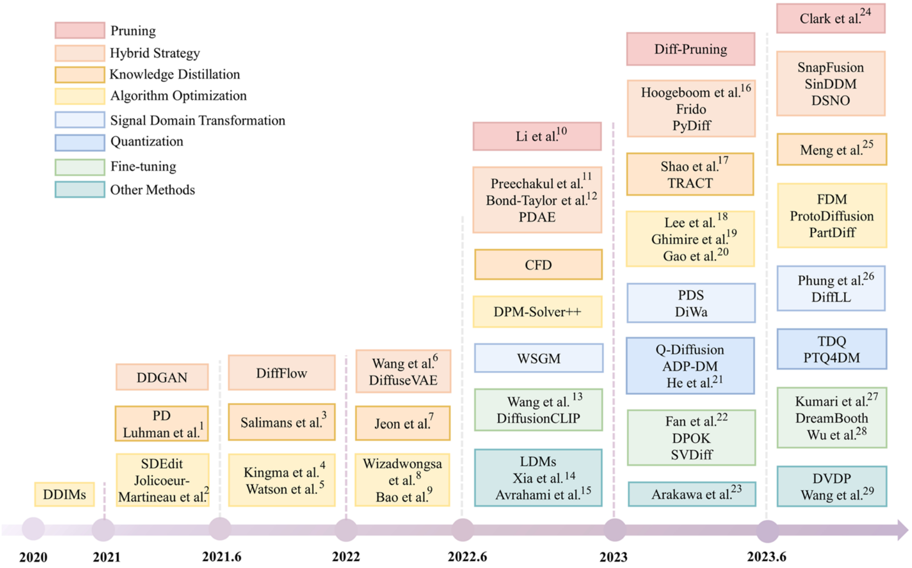


Fig. 20 A roadmap of lightweight DMs methods summarized in chronological order. Different colored boxes in the figure represent different categories of methods. Representative models in each category are shown in the boxes. The corresponding relationship between superscripts and papers is: 1. (Luhman and Luhman 2021), 2. (Jolicoeur-Martineau et al. 2021), 3. (Salimans and Ho 2021), 4. (Kingma et al. 2021), 5. (Watson et al. 2021), 6. (Wang et al. 2022c), 7. (Jeon and Park 2022), 8. (Wizadwongsa and Suwajanakorn 2022), 9. (Bao et al. 2022), 10. (Li et al. 2022b), 11. (Preechakul et al. 2022), 12. (Bond-Taylor et al. 2022), 13. (Wang et al. 2022b), 14. (Xia et al. 2022), 15. (Avrahami et al. 2023), 16. (Hoogeboom et al. 2023), 17. (Shao et al. 2023), 18. (Lee et al. 2023b), 19. (Ghimire et al. 2023), 20. (Gao et al. 2023b), 21. (He et al. 2023a), 22. (Fan and Lee 2023), 23. (Arakawa et al. 2023), 24. (Clark and Jaini 2023), 25. (Meng et al. 2023), 26. (Phung et al. 2023), 27. (Kumari et al. 2023), 28. (Wu et al. 2023), 29. (Wang et al. 2023a). (Color figure online)

Table 1 Classification table of methods for lightweight DMs

Lightweight method	Characteristic	Task	Model
Knowledge distillation: the main idea of lightweight is to compress the knowledge of the teacher model into the student model, so that the number of sampling steps of student DMs is greatly reduced	1. Multiple distillations are allowed for smaller number of steps, according to application requirements 2. The sampling steps of large pre-trained DMs are allowed to be further reduced	Image generation	CD (Song et al. 2023)[a], PD (Luhman and Luhman 2021)[a], (Salimans and Ho 2021)[a], (Meng et al. 2023)[a] CFD (Sun et al. 2022), TRACT (Berthelot et al. 2023)[a]
		Image-to-image translation	(Meng et al. 2023)[a]
		Image inpainting	CD (Song et al. 2023)[a], (Meng et al. 2023)[a]
		Super-resolution	CD (Song et al. 2023)[a]
		Colorization	CD (Song et al. 2023)[a]
		Image synthesis	CUD (Shao et al. 2023)[a]
Quantization: the main idea of lightweight is to sacrifice a certain degree of accuracy of DMs to reduce the cost of single network evaluation	1. A trade-off method between the accuracy and cost of DMs is provided 2. Large pre-trained DMs are allowed to further reduce storage costs	Image generation	TDQ (So et al. 2023)[a], ADP-DM (Wang et al. 2023b)[a], PTQ4DM (Shang et al. 2023b)[a], (He et al. 2023a)
		Text-to-image	Q-Diffusion (Li et al. 2023a)[a]
Pruning: the main idea of lightweight is to reduce redundant or insignificant computing costs in a single network evaluation by streamlining DMs	1. The size of large pre-trained DMs can be reduced 2. Unimportant calculations are removed, making the evaluation network for DMs leaner and easier to understand 3. The pruning ratio of DMs needs to be weighed. Different degrees of pruning have different proportions of influence on lightweight effect and result quality	Image generation	Diff-Pruning (Fang et al. 2023b)[a]
		Image editing	Li et al. (2022b)
		Image classification	Clark and Jain i(2023)[a]

Table 1 (continued)

Lightweight method	Characteristic	Task	Model
Fine-tuning: the main idea of lightweight is to fine-tune the pre-trained DMs for specific tasks, greatly reducing the cost of retraining	1. Large pre-trained DMs can quickly adapt to new tasks with a small number of sample data. Fine-tuning DMs can learn more specific, task-relevant features in fewer iterations than training from scratch 2. Fine-tuning pre-trained DMs can reduce the overfitting phenomenon on the target task	Text-to-image	DiffusionCLIP (Kim et al. 2022b)[a], DPOK (Fan et al. 2023b)[a]
			SVDiff (Han et al. 2023)[a], DreamBooth (Ruiz et al. 2023a)[a]
			Kumari et al. (2023)[a], Jia et al. (2023)[a], DomainStudio (Zhu et al. 2023)
		Image-to-image translation	Wang et al. (2022b)
		Image generation	Fan and Lee (2023)[a], Xie et al. (2023)[a]
		Image editing	Wu et al. (2023a)[a]
Signal domain transformation: the main idea of lightweight is that different frequency components are processed in a targeted manner in the denoising stage of DMs to reduce computational costs	1. The processing of different frequency content in the image is weighed to allocate computing resources 2. Reduce the dimension of features processed by DMs and improve the compactness of data	Image generation	WSGM (Guth et al. 2022), (Phung et al. 2023), PDS (Zhang et al. 2023c)[a]
		Super-resolution	DiWa (Moser et al. 2023)
		Low-light image enhancement	DiffLL (Jiang et al. 2023b)

Table 1 (continued)

Lightweight method	Characteristic	Task	Model
Algorithm optimization: the main idea of lightweight is to optimize the training strategy or sampler of DMs by introducing interpretable principles or new thinking angles	1. By focusing on the characteristics of DMs, targeted lightweight directions such as selective calculation, modification of sampling trajectories, and skipping of some steps can be developed 2. It is necessary to use reasonable analysis and mathematical derivation to enhance the adaptability of new algorithms to DMs 3. Part of the lightweight methods for sampling can reduce the number of sampling steps for pre-trained DMs without additional training	Image generation	DDIMs (Song et al. 2020b), (Ghimire et al. 2023), ReDi (Zhang et al. 2023e)[a], (Hang et al. 2023), (Gao et al. 2023b)[a], UniPC (Zhao et al. 2023a)[a], (Golnari et al. 2023)[a], DDPM-IP (Ning et al. 2023), (Lee et al. 2023b), (Permenter and Yuan 2023)[a], (Haxholli and Lorenzi 2023), ProtoDiffusion (Baykal et al. 2023)
		Image editing	SDEdit (Meng et al. 2021)[a], (Li et al. 2022b)[a]
		Image synthesis	SDEdit (Meng et al. 2021)[a], DDM (Huang et al. 2023a), FDM (Wu et al. 2023b)
		Image denoising	CoreDiff (Gao et al. 2023a)
		Super-resolution	ResShift (Yue et al. 2023), (Wizadwongsa and Suwajanakorn 2022)[a], PartDiff (Zhao et al. 2023b)
			DDM (Huang et al. 2023a), ACDMSR (Niu et al. 2023)[a]
		Text-to-image	DPM-Solver++ (Lu et al. 2022b)[a], (Wizadwongsa and Suwajanakorn 2022)[a], DVAR (Voronov et al. 2023)
			Permenter and Yuan (2023)[a], OMS-DPM (Liu et al. 2023b)[a]
		Image inpainting	Wizadwongsa and Suwajanakorn (2022)[a]
		Colorization	Wizadwongsa and Suwajanakorn (2022)[a]

Table 1 (continued)

Lightweight method	Characteristic	Task	Model
Hybrid strategy: the main idea of lightweight is to combine DMs with other efficient network architectures or technologies to improve the comprehensive capabilities of DMs and further reduce costs	1. The advantages of each method are used to make up for the shortcomings of DMs, making the lightweight model more flexible and adaptable 2. In the process of combining with DMs, it is necessary to consider whether to add too much extra memory, the number of parameters, or the complexity of the model, so as to improve the compatibility of DMs with various application scenarios	Image generation	SinDDM (Kulikov et al. 2023)[a], DSNO (Zheng et al. 2023b)[a], (Bond-Taylor et al. 2022)[a], (Preechakul et al. 2022) DiffFlow (Zhang and Chen 2021),INDM (Kim et al. 2022a)[a], DDGAN (Xiao et al. 2021), (Wang et al. 2022c), PDDPM (Ryu and Ye 2022), ToMe (Bolya and Hoffman 2023), SSIDMs (Xu et al. 2023), MEME (Lee et al. 2023a), (Zheng et al. 2023a)[a]
		Text-to-image	(Zhang and Agrawala 2023), SnapFusion (Li et al. 2023b; Hoogeboom et al. 2023), (Xiao et al. 2023a)[a], BK-SDMs (Kim et al. 2023)[a], SSIDMs (Xu et al. 2023), (Ruiz et al. 2023b)[a],
		Image synthesis	DiffuseVAE (Pandey et al. 2022), Frido (Fan et al. 2023), (Leng et al. 2023)
		Image reconstruction	PDAE (Zhang et al. 2022a)[a], LDIR (He et al. 2023d)[a]
		Super-resolution	ResDiff (Shang et al. 2023a),LDDPM (Wang et al. 2022a)
		Image inpainting	PSM (Li et al. 2022d)
		Low-light image enhancement	PyDiff (Zhou et al. 2023b)
		Image restoration	DiffIR (Xia et al. 2023b)
		Image editing	LEDITS (Tsaban and Passos 2023)[a]

Table 1 (continued)

Lightweight method	Characteristic	Task	Model
Others: in specific modifications, more various methods are allowed to be used	Lightweight approaches can optimize DMs in several ways. These include reducing the cost of a single evaluation of the network, the number of times the network is used, the cost of model training, and storage requirements	Image synthesis	DVDP (Zhang et al. 2023b), LDMs (Rombach et al. 2022)
		Image generation	Arakawa et al. (2023), Wang et al. (2023a)
		Image reconstruction	Xia et al. (2022)
		Image editing	Avrahami et al. (2023)[a]

A summary of representative methods for lightweight DMs under each category

[a] The pre-trained DMs can be used

provides a compact discrete image representation that can be modeled more efficiently than pixel or continuous latent representation. This binary latent DMs can have higher sampling efficiency and representation ability of HR images.

The above is a detailed introduction to the current methods of lightweight DMs in the field of image processing. Figure 20 summarizes the development of different categories of methods. To quickly understand the key points and related work of each lightweight method, a classification table (Table 1) is summarized. The first column of the table contains a brief description of each method, and the second column highlights the characteristics of the corresponding method. In addition, in order to facilitate researchers to quickly find methods for specific tasks, the third and fourth columns divide relevant papers according to task types.

To better demonstrate the performance of the models, the experimental results of image generation of lightweight DMs are summarized, involving the performance on CIFAR10 (Krizhevsky et al. 2009) and LSUN-Bedroom datasets (Yu et al. 2015)(Table 2). The table contains the lightweight type "Lightweight Method" of the model "Model", and the values of the Number of Function Evaluations (NFE), Fréchet Inception Distance (FID) (Heusel et al. 2017) and Inception Score (IS) (Salimans et al. 2016) indexes on the two datasets. The model parameters "Params" and hardware configuration "Hardware" are also provided. Because some papers do not show relevant results, they are not listed here.

4 Conclusion and prospect

DMs have become a hot topic in the field of image processing due to their competitive performance and great potential. However, their high computing and storage costs are not friendly to researchers who lack high performance hardware equipment, thus limiting their application. For applications that need to be built on intelligent edge devices, it is more necessary to design lightweight methods. To reduce the cost of training and inference stages while generating high-quality samples, many methods have been proposed in recent years. These methods further lighten DMs from KD, quantization, pruning, fine-tuning, signal domain transformation, algorithm optimization, hybrid strategy, and other different perspectives. This survey first introduces the basic principle of DMs, and then sorts out these types of methods. However, current lightweight approaches to DMs have not been fully explored, implying that there are still issues that require attention. This section discusses these issues and provides insights into future prospects.

4.1 Theoretical derivation

DMs are derived from concepts in physics and are supported by interpretable mathematical theories. By optimizing the training strategies (Hang et al. 2023; Lee et al. 2023b) and samplers (Gao et al. 2023b; Golnari et al. 2023) of DMs from the perspective of theoretical derivation, some results have been achieved in improving training speed and reducing sampling costs. However, these basic principles still need to be studied in depth, and there is still huge room for theoretical expansion in lightweight DMs. Some possible research directions include:

(1) Relevant theories for adjusting the diffusion process

The sampling of DMs is equivalent to solving the corresponding SDE or ODE (Zhou et al. 2023a; Guo et al. 2023). Therefore, it is necessary to reason about efficient and

Table 2 An experimental summary of lightweight DMs on image generation tasks

Model	Lightweight method	CIFAR10(32 × 32)			LSUN-Bedroom(256 × 256)			Params.	Hardware
		NFE↓	FID↓	IS↑	NFE↓	FID↓	IS↑		
Salimans and Ho (2021)	Knowledge distillation	1	9.12	–	–	–	–	60.0M	8 × /64 × TPUv4
CUD (Shao et al. 2023)		1	3.37	9.42	–	–	–	–	4 × A100
CD (Song et al. 2023)		2	2.93	9.75	–	–	–	–	–
TRACT (Berthelot et al. 2023)		2	3.32	–	–	–	–	60 M	8 × A100
PD (Luhman and Luhman 2021)		1	9.36	8.36	–	–	–	35.7M	–
PTQ4DM (Shang et al. 2023b)	Quantization	100	5.69	8.82	100	7.48	2.23	–	–
TDQ (So et al. 2023)		100	5.99	8.85	–	–	–	–	4 × A100 and 8 × 3090
ADP-DM (Wang et al. 2023b)		100	4.24	9.07	100	6.46	2.55	–	–
He et al. (2023a)		–	–	–	200	3.75	sFID↓9.89	– -	–
Q-Diffusion (Li et al. 2023a)		100	4.24	9.42	200	3.80	sFID↓9.95	–	–
Diff-Pruning (Fang et al. 2023b)	Pruning	100	5.29	–	100	18.6	–	19.8M/46.5M	–
Fan and Lee (2023)	Fine-tuning	10	2.59	–	–	–	–	–	4 × 2080 Ti
PDS (Zhang et al. 2023c)	Signal domain Transformation	1000	1.99	–	–	–	–	–	1 × 3090
Phung et al. (2023)		4	4.01	–	–	–	–	–	1–8 × A100
DDIMs (Song et al. 2020a)	Algorithm optimization	50	4.67	8.78	100	6.62	–	35.7M	1 × 2080 Ti
Gao et al. (2023b)		–	–	–	8	27.18	–	–	–
fastDPM (Kong and Ping (2021)		10	9.90	–				35.7M	–
Kingma et al. (2021)		1000	2.67	–	–	–	–	–	–
Jolicoeur-Martineau et al. (2021)		180	2.44	–	–	–	–	–	–
Bao et al. (2022)		10	12.19	–	–	–	–	52.6M	–
Lu et al. (2022b)		5	29.22	–	5	17.79	–	–	–
Permenter and Yuan (2023)		10	3.85	–	–	–	–	–	1×4090
UniPC (Zhao et al. 2023a)		10	3.87	–	5	11.88	–	–	–
Haxholli and Lorenzi (2023)		1000	8.89	–	–	–	–	–	1×2080

Table 2 (continued)

Model	Lightweight method	CIFAR10(32 × 32)			LSUN-Bedroom(256 × 256)			Params.	Hardware
		NFE↓	FID↓	IS↑	NFE↓	FID↓	IS↑		
Baykal et al. (2023)		–	8.55	9.01	–	–	–	–	4×A100
DDPM-IP (Ning et al. (2023)		100	3.12	sFID↓3.86	–	–	–	57 M	2×V100
Lee et al. (2023b)		118	2.45	9.55	–	–	–	57.93M	–
DDGAN (Xiao et al. 2021)	Hybrid strategy	4	3.75	9.63	–	–	–	–	4×V100
DSNO (Zheng et al. 2023b)		1	3.78	–	–	–	–	65.8M	–
Bond-Taylor et al. (2022)		–	–	–	–	3.64	Recall↑0.38	145 M	1×2080 Ti
DiffFlow (Zhang and Chen 2021)		100	14.14	–	–	–	–	≈36 M	–
INDM (Kim et al. 2022a)		1000	2.28	–	–	–	–	118 M	–
Preechakul et al. (2022)*		–	–	–	100	5.70	–	–	–
Wang et al. (2022c)		–	3.19	Recall↑0.58	100	3.65	Recall↑0.32	–	4×/8×V100
Arakawa et al. (2023)*	Other	–	–	–	1000	24.1	–	–	–
Wang et al. (2023a)		–	–	–	64	3.85	–	–	V100 and 3090

The table summarizes the experimental data of lightweight DMs performing image generation tasks on CIFAR10 and LSUN-Bedroom datasets. The order in the table does not represent the performance ranking of the models. Some of the IS are filled with the results of Spatial Fréchet Inception Distance (sFID) (Nash et al. 2021) and Recall (Kynkäänniemi et al. 2019) indexes, which have been noted in the table

"↑" means that the larger the index value, the better, and "↓" has the opposite meaning. Unpublished experimental data are represented by "–". Contents with a "/" symbol indicate that the former is applicable to the CIFAR10 dataset and the latter is applicable to the LSUN-Bedroom dataset. Models with "*" use the LSUN-Bedroom dataset of size 128 × 128. The quantization models in the table all use 8-bit weights and 8-bit activation quantization. The "Params" of Salimans and Ho (2021), Bao et al. (2022), Zhang and Chen (2021), Kong and Ping (2021) are cited from Zheng et al. (2023b). All index values for Shang et al. (2023b) and Lu et al. (2022b) are cited from Wang et al. (2023b) and Zhao et al. (2023a) respectively. The index values for LSUN-Bedroom dataset of Li et al. (2023a) and Song et al. (2020a) are cited from He et al. (2023a) and Zheng et al. (2023b) respectively. The index values for CIFAR10 of Song et al. (2020a) are cited from Xiao et al. (2021). The remaining data are cited from their respective papers accurate samplers. For example, the current sampler will cause certain errors by reducing the number of solutions. Zhou et al. improved the ODE-based sampler by directly learning the mean direction to eliminate truncation errors (Zhou et al. 2023a). Guo et al. proposed a new Gaussian mixture solver based on SDE by relaxing the Gaussian reverse kernel assumption (Guo et al. 2023). Xia et al. adjusts the network by aligning the sampling distribution with the real distribution and obtaining new time steps to find a more accurate integral direction (Xia et al. 2023a). In addition, the performance of the sampler can be improved by reparameterizing the diffusion process (Zhang et al. 2023a) or mapping it to a more amenable space for sampling (Pandey et al. 2023). This ensures that results are generated in smaller steps. Using theoretical derivation, obtaining samplers that can generate new samples without training is also worthy of attention (Scarvelis et al. 2023).

Reasoning about and adjusting more accurate training targets also helps improve training efficiency. The general constant loss weight strategy in DMs will lead to biased estimation during the training phase. For example, Yu et al. proposed an effective weighting strategy based on the theoretical unbiased principle (Yu et al. 2023a). Kang et al. used observations following the Bernoulli distribution to concretize a surrogate loss for negative log-likelihood (Kang et al. 2023). This realigns training goals to trade off quality versus speed. The assumption that the original goals of DMs are invariant to noise processes may also affect model performance. For example, this assumption was eliminated as Sekhar et al (Sekhar Sahoo et al. 2023). They use Bayesian inference to adjust the noise schedule based on the specific characteristics of each image instance. Lin et al. used noise schedules with zero terminal SNR to ensure that training behavior is aligned with inference (Lin et al. 2024).

(2) Improve the theoretical framework of DMs from different view

By deriving it from multiple different perspectives, it helps to gain a deeper understanding of the theoretical basis of DMs. There is already work devoted to optimization from a physics perspective. For example, Ambrogioni derived through the perspective of equilibrium statistical mechanics (Ambrogioni 2023). And Corso et al. used a time-evolving potential to guide the inference process (Corso et al. 2023). In addition, introducing concepts from other fields is also considered as a potential direction. For example, by incorporating RL (Fan et al. 2023b) or consistency training theory (Song et al. 2023; Song and Dhariwal 2023; Kim et al. 2023) into DMs for further derivation to improve training strategies.

In addition to the above improvement directions, DMs for different image tasks also need to improve their interpretability accordingly. In the context of specific tasks, such as IR, SR, etc., there is still the problem of insufficient theoretical elaboration of the significant performance of DMs (Nie et al. 2023).

4.2 Architecture design

The efficiency and application cost of DMs are closely related to the design of network architecture. Most current research is based on the UNet architecture, and vanilla UNet used in DMs usually requires a large amount of computing resources (Xiao et al. 2023b). Therefore, further research is needed to gain a deeper understanding of the capabilities and limitations of different modules in DMs networks. For example, Huang et al. found that UNet is often affected by unstable training in DMs, and reducing its long skip connection coefficient can alleviate this problem (Huang et al. 2023b). Therefore, they proposed an efficient coefficient scaling framework to ensure training stability and improve speed. Xiao et al. focused on simplifying channel attention and introducing simple gate operations (Xiao et al. 2023b). And Zheng et al. introduced LEGO bricks for effective stacking (Zheng et al. 2023c). They all succeeded in achieving a more lightweight network while maintaining good predictive performance. In addition, modules designed for specific tasks can simplify calculations. For example, the Prompt-Aware Introverted Attention layer designed by Manukyan et al (Manukyan et al. 2023). This module improves the self-attention score, thereby producing better TTI results. Xiao et al. added a conditional prior enhancement module to help obtain prior information to assist SR tasks (Xiao et al. 2023b).

Integrating other efficient and mature network architectures, such as GANs (Xiao et al. 2021), AE (Pandey et al. 2022), etc. for lightweighting also has obvious advantages. Existing works have demonstrated that more effective and faster models can be obtained by combining different complementary network architectures. Currently, only a limited part of the attributes are manipulated after combining different works (Leng et al. 2023). Therefore, future research needs to be adjusted in practical applications and focus on the respective strengths and weaknesses of the combined network architectures. The extent of their influence in forward and reverse processes and in different tasks should be studied (Lee et al. 2023a). For example, PDAE first studies the modeling characteristics of DMs (Zhang et al. 2022a). After that, meaningful partial information is targetedly generated by AE, which is good at representation learning. The rest are rebuilt by DMs. This improves the training efficiency and performance of DMs. Xia et al. assigned the task of predictive priori representation and prediction high-quality results according to the advantages of each network to reduce the impact of errors (Xia et al. 2023b). Ultimately the method produces more stable and realistic results with fewer iterations. At the same time, the scheduling of different models is also important (Liu et al. 2023b; Tsaban and Passos 2023). This allows the advantages of DMs in training stability and mode coverage to be better exploited. And the capabilities of each network are better integrated. Lightweight DMs of higher quality and computational efficiency can be constructed for desired scenarios.

4.3 Technology integration

Model compression techniques, such as KD (Salimans and Ho 2021; Shao et al. 2023), quantization (Song et al. 2020a; So et al. 2023), pruning (Fang et al. 2023b), and fine-tuning (Kumari et al. 2023; Ruiz et al. 2023a), are widely recognized as effective lightweight approaches. However, further in-depth research is needed in adapting the method of compressing DMs for different needs. As Starodubcev et al. observed that when distilling DMs for TTI, the student model outperformed the teacher model in some sample generation (Starodubcev et al. 2023). Therefore, they proposed an adaptive collaborative pipeline to assist distillation. In addition to improvements in task customization, careful design of different network modules (Lee et al. 2023d) and sampling stages (Wang et al. 2023c; Mei et al. 2023; Yin et al. 2023) can help improve the suitability of compression techniques. For example, the network is pruned into different scales according to different generation steps to avoid redundant calculations (Yang et al. 2023a). As a technology that introduced lightweight DMs earlier, distillation has achieved faster development than other compression technologies. However, introducing other compression techniques can also help achieve more competitive results. Combined with pruning methods, Lottery Ticket Hypothesis can find sparser sub-models, thereby reducing storage and computing pressure (Jiang et al.

2023a). Methods of superimposing multiple technologies, including integration between compression technologies (Chang et al. 2023; He et al. 2023b) and integration with other architectures (Sauer et al. 2023), are all directions that can be explored in depth in future research.

In addition, pre-trained DMs have been extensively learned on large-scale data, which can capture rich semantic information. To make full use of these existing knowledge and achieve the effect of multi-purpose models, optimization techniques that utilize pre-trained DMs are attractive (Clark and Jaini 2023; Ruiz et al. 2023a; Laousy et al. 2023). However, pre-trained models may face the problem of catastrophic forgetting when performing downstream tasks. At the same time, in the case of limited data, the research on few-shot (Ruiz et al. 2023a; Masip et al. 2023) or zero-shot (Clark and Jaini 2023; Kim et al. 2022b; He et al. 2023d; Nguyen and Tran 2023) has become a direction worthy of attention. This may enable DMs to generate satisfactory results in unseen domains and even in other tasks. Although good results have been achieved in this field, most of the work is still limited by the capabilities of pre-trained DMs (Wang et al. 2022b; Han et al. 2023; Ruiz et al. 2023a; Permenter and Yuan 2023) or weakens the capabilities of the original model (Lu et al. 2023). Therefore, future research may need to resort to means such as more powerful prior conditions (Ma et al. 2023a) or semantic information to more effectively share the knowledge of pre-trained DMs. This will help tap the potential of large-scale DMs and bring more possibilities for model optimization.

4.4 Multi-domain processing

The image generally has redundancy in space, and the sampling of DMs needs to be processed in the same dimension. This results in a large computational and storage overhead. Images can be transformed from the pixel domain to the frequency domain for processing to solve this problem. For example, Phung et al. (2023) and Jiang et al. (2023b) extract low-frequency and high-frequency components from the images through DWT, and perform adaptive processing respectively. This processing method can not only capture information on different frequencies, but also improve the efficiency of data expression. However, it is worth noting that the number of studies in the direction of frequency domain processing is currently lacking, and further research is needed.

The exploration of the latent space domain has also received more and more attention. Rombach et al. (2022) uses AE to convert image data from pixel space to latent space for processing. This method of reducing data dimensions has been proven to effectively improve data compactness and reduce computational complexity. There is also related work in the follow-up, but the process of transforming the latent space requires an additional encoder, which adds additional complexity to the diffusion framework (Hoogeboom et al. 2023). However, there may be cases where other pre-trained encoders are not suitable for DMs processing (Wang et al. 2023a). Therefore, this aspect is still worthy of further exploration, so that future DMs can reduce the dimensionality and complexity of the data to a greater extent while maintaining the semantic information of the data.

In addition, the latent space domain is easier to integrate with other domains, providing potential opportunities for lightweight DMs. For example, the features processed by LDMs in the latent space are then segmented based on patches to further reduce the feature scale (Ma et al. 2023b). Luo et al. successfully enhanced the frequency component from latent space to pixel space by introducing a frequency compensation module (Luo et al. 2023a). It is worth noting that in the latest research, the consistency model (Song et al. 2023) is also transformed into the latent space, and the inference process is further accelerated through fine-tuning (Luo et al. 2023b, c). Therefore, future in-depth research on the characteristics and advantages of latent space is expected to bring more possibilities to lightweight DMs.

It is hoped that the efforts of this work can further promote the interest of scholars in various fields to study lightweight DMs. More efficient, reliable and user-friendliness lightweight DMs solutions can be provided in the future. And it is expected that the potential of DMs will break through the current limitations and play their role in real-time and resource-limited application scenarios.

Author contributions All authors contributed to the conception of the study; Material collection was done by M.Z., Y.Z. and X.Z.; Arrangement of materials was done by W.M. and W.S.; The first draft of the manuscript was written by W.M.; W.S. and W.M. reviewed and advised this article; All authors read and approved the final manuscript.

Funding This work was supported by the National Natural Science Foundation of China (Grant numbers 52071349) and Independent Research Projects for Postgraduate Students (Grant numbers SZKY2023054).

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

References

Ambrogioni L (2023) The statistical thermodynamics of generative diffusion models. Preprint. arXiv:2310.17467

Arakawa S, Tsunashima H, Horita D, Tanaka K, Morishima S (2023) Memory efficient diffusion probabilistic models via patch-based generation. Preprint. arXiv:2304.07087

Avrahami O, Lischinski D, Fried O (2022) Blended diffusion for text-driven editing of natural images. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 18208–18218

Avrahami O, Fried O, Lischinski D (2023) Blended latent diffusion. ACM Trans Graph (TOG) 42(4):1–11

Bai H, Hou L, Shang L, Jiang X, King I, Lyu MR (2022) Towards efficient post-training quantization of pre-trained language models. In: Advances in neural information processing systems, vol 35, pp 1405–1418

Balaji Y, Nah S, Huang X, Vahdat A, Song J, Kreis K, Aittala M, Aila T, Laine S, Catanzaro B et al (2022) ediffi: text-to-image diffusion models with an ensemble of expert denoisers. Preprint. arXiv:2211.01324

Bao F, Li C, Sun J, Zhu J, Zhang B (2022) Estimating the optimal covariance with imperfect mean in diffusion probabilistic models. In: International conference on machine learning, pp 1555–1584. PMLR

Bao F, Nie S, Xue K, Cao Y, Li C, Su H, Zhu J (2023) All are worth words: a vit backbone for diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 22669–22679

Baykal G, Karagoz H.F, Binhuraib T, Unal G (2023) ProtoDiffusion: classifier-free diffusion guidance with prototype learning. Preprint. arXiv:2307.01924

Berthelot D, Autef A, Lin J, Yap DA, Zhai S, Hu S, Zheng D, Talbot W, Gu E (2023) TRACT: denoising diffusion models with transitive closure time-distillation. Preprint. arXiv:2303.04248

Blattmann A, Rombach R, Ling H, Dockhorn T, Kim SW, Fidler S, Kreis K (2023) Align your latents: high-resolution video synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 22563–22575

Bolya D, Hoffman J (2023) Token merging for fast stable diffusion. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 4598–4602

Bond-Taylor S, Hessey P, Sasaki H, Breckon TP, Willcocks CG (2022) Unleashing transformers: parallel token prediction with discrete absorbing diffusion for fast high-resolution image generation from vector-quantized codes. In: European conference on computer vision. Springer, Cham, pp 170–188

Chang H, Shen H, Cai Y, Ye X, Xu Z, Cheng W, Lv K, Zhang W, Lu Y, Guo H (2023) Effective Quantization for Diffusion Models on CPUs. Preprint. arXiv:2311.16133

Cheikh Tourad EH (2022) Eleuldj M Quantization and pipelined hardware implementation of deep neural network models. J Comput Sci 18(11):1021–1029. https://doi.org/10.3844/jcssp.2022.1021.1029

Chen N, Zhang Y, Zen H, Weiss R.J, Norouzi M, Chan W (2020) WaveGrad: estimating gradients for waveform generation. In: International Conference on Learning Representations

Chen D, Mei J-P, Zhang H, Wang C, Feng Y, Chen C (2022a) Knowledge distillation with the reused teacher classifier. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 11933–11942

Chen Z, Wu Y, Leng Y, Chen J, Liu H, Tan X, Cui Y, Wang K, He L, Zhao S et al (2022b) Resgrad: residual denoising diffusion probabilistic models for text to speech. Preprint. arXiv:2212.14518

Chin T-W, Ding R, Zhang C, Marculescu D (2020) Towards efficient model compression via learned global ranking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1518–1528

Church KW, Chen Z, Ma Y (2021) Emerging trends: a gentle introduction to fine-tuning. Nat Lang Eng 27(6):763–778

Clark K, Jaini P (2023) Text-to-image diffusion models are zero-shot classifiers. In: ICLR 2023 workshop on mathematical and empirical understanding of foundation models

Corso G, Xu Y, De Bortoli V, Barzilay R, Jaakkola T (2023) Particle guidance: non-IID diverse sampling with diffusion models. In: NeurIPS 2023 workshop on deep learning and inverse problems

Dhariwal P, Nichol A (2021) Diffusion models beat gans on image synthesis. In: Advances in neural information processing systems, vol 34, pp 8780–8794

Dockhorn T, Vahdat A (2022) Genie: higher-order denoising diffusion solvers. In: Advances in neural information processing systems, vol 35, pp 30150–30166

Dockhorn T, Vahdat A, Kreis K (2021) Score-based generative modeling with critically-damped langevin diffusion. In: International conference on learning representations

Esser P, Rombach R, Ommer B (2021) Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 12873–12883

Fan Y, Lee K (2023) Optimizing ddpm sampling with shortcut fine-tuning. Preprint. arXiv:2301.13362

Fan W-C, Chen Y-C, Chen D, Cheng Y, Yuan L, Wang Y-CF (2023) Frido: feature pyramid diffusion for complex scene image synthesis. In: Proceedings of the AAAI conference on artificial intelligence, vol 37, pp 579–587

Fan Y, Watkins O, Du Y, Liu H, Ryu M, Boutilier C, Abbeel P, Ghavamzadeh M, Lee K, Lee K (2023) DPOK: reinforcement learning for fine-tuning text-to-image diffusion models. Preprint. arXiv:2305.16381

Fang G, Ma X, Song M, Mi MB, Wang X (2023a) DepGraph: towards any structural pruning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, p 16091–16101

Fang G, Ma X, Wang X (2023b) Structural pruning for diffusion models. Preprint. arXiv:2305.10924

Gao Q, Li Z, Zhang J, Zhang Y, Shan H (2023a) CoreDiff: contextual error-modulated generalized diffusion model for low-dose CT denoising and generalization. Preprint. arXiv:2304.01814

Gao Y, Pan Z, Zhou X, Kang L, Chaudhari P (2023b) Fast diffusion probabilistic model sampling through the lens of backward error analysis. Preprint. arXiv:2304.11446

Ghimire S, Liu J, Comas A, Hill D, Masoomi A, Camps O, Dy J (2023) Geometry of score based generative models. Preprint. arXiv:2302.04411

Go H, Lee Y, Kim J-Y, Lee S, Jeong M, Lee HS, Choi S (2023) Towards practical plug-and-play diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1962–1971

Golnari PA, Yao Z, He Y (2023) Selective guidance: are all the denoising steps of guided diffusion important? Preprint. arXiv:2305.09847

Gong S, Li M, Feng J, Wu Z, Kong L (2022) Diffuseq: sequence to sequence text generation with diffusion models. In: The 11th International conference on learning representations

Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: Proceedings of the 27th international conference on neural information processing systems, vol 2, pp 2672–2680

Gu S, Chen D, Bao J, Wen F, Zhang B, Chen D, Yuan L, Guo B (2022) Vector quantized diffusion model for text-to-image synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10696–10706

Gui J, Sun Z, Wen Y, Tao D, Ye J (2023) A review on generative adversarial networks: algorithms, theory, and applications. IEEE Trans Knowl Data Eng 35(4):3313–3332

Guo HA, Lu C, Bao F, Pang T, Shuicheng Y, Du C, Li C (2023) Gaussian mixture solvers for diffusion models. In: 37th Conference on neural information processing systems

Guth F, Coste S, De Bortoli V, Mallat S (2022) Wavelet score-based generative modeling. In: Advances in neural information processing systems, vol 35, pp 478–491

Han L, Li Y, Zhang H, Milanfar P, Metaxas D, Yang F (2023) SVDiff: compact parameter space for diffusion fine-tuning. Preprint. arXiv:2303.11305

Hang T, Gu S, Li C, Bao J, Chen D, Hu H, Geng X, Guo B (2023) Efficient diffusion training via Min-SNR weighting strategy. Preprint. arXiv:2303.09556

Haxholli E, Lorenzi M (2023) Faster training of diffusion models and improved density estimation via parallel score matching. Preprint. arXiv:2306.02658

He Y, Liu L, Liu J, Wu W, Zhou H, Zhuang B (2023a) PTQD: accurate post-training quantization for diffusion models. Preprint. arXiv:2305.10657

He Y, Liu J, Wu W, Zhou H, Zhuang B (2023b) EfficientDM: efficient quantization-aware fine-tuning of low-bit diffusion models. Preprint. arXiv:2310.03270

He J, Liu J, Ye Z, Huang R, Cui C, Liu H, Zhao Z (2023c) RMSSinger: realistic-music-score based singing voice synthesis. Preprint. arXiv:2305.10686

He L, Yan H, Luo M, Luo K, Wang W, Du W, Chen H, Yang H, Zhang Y (2023d) Iterative reconstruction based on latent diffusion model for sparse data reconstruction. Preprint. arXiv:2307.12070

Heusel M, Ramsauer H, Unterthiner T, Nessler B, Hochreiter S (2017) Gans trained by a two time-scale update rule converge to a local Nash equilibrium. In: Advances in neural information processing systems, vol 30, pp 6626–6637

Ho J, Salimans T (2021) Classifier-free diffusion guidance. In: NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications

Ho J, Jain A, Abbeel P (2020) Denoising diffusion probabilistic models. In: Advances in neural information processing systems, vol 33, pp 6840–6851

Ho J, Saharia C, Chan W, Fleet DJ, Norouzi M (2022) Salimans T Cascaded diffusion models for high fidelity image generation. J Mach Learn Res 23(1):2249–2281

Hoogeboom E, Heek J, Salimans T (2023) simple diffusion: end-to-end diffusion for high resolution images. Preprint. arXiv:2301.11093

Hu E.J, Wallis P, Allen-Zhu Z, Li Y, Wang S, Wang L, Chen W et al (2021) Lora: low-rank adaptation of large language models. In: International conference on learning representations

Huang T, You S, Wang F, Qian C, Xu C (2022) Knowledge distillation from a stronger teacher. In: Advances in neural information processing systems, vol 35, pp 33716–33727

Huang Y, Qin Z, Liu X, Xu K (2023a) Decoupled diffusion models with explicit transition probability. Preprint. arXiv:2306.13720

Huang Z, Zhou P, Shuicheng Y, Lin L (2023b) Scalelong: towards more stable training of diffusion model via scaling network long skip connection. In: 37th Conference on neural information processing systems

Jeon J, Park N (2022) SPI-GAN: distilling score-based generative models with straight-path interpolations. Preprint. arXiv:2206.14464

Jia X, Zhao Y, Chan KC, Li Y, Zhang H, Gong B, Hou T, Wang H, Su Y-C (2023) Taming encoder for zero fine-tuning image customization with text-to-image diffusion models. Preprint. arXiv:2304.02642

Jiang C, Hui B, Liu B, Yan D (2023a) Successfully applying lottery ticket hypothesis to diffusion model. Preprint. arXiv:2310.18823

Jiang H, Luo A, Han S, Fan H, Liu S (2023b) Low-light image enhancement with wavelet-based diffusion models. Preprint. arXiv:2306.00306

Jolicoeur-Martineau A, Li K, Piché-Taillefer R, Kachman T, Mitliagkas I (2021) Gotta go fast when generating data with score-based models. Preprint. arXiv:2105.14080

Kang J, Choi J, Choi S, Han B (2023) Observation-guided diffusion probabilistic models. Preprint. arXiv:2310.04041

Karras T, Aittala M, Aila T, Laine S (2022) Elucidating the design space of diffusion-based generative models. In: Advances in neural information processing systems, vol 35, pp 26565–26577

Kim B-K, Song H-K, Castells T, Choi S (2023) On architectural compression of text-to-image diffusion models. Preprint. arXiv:2305.15798

Kim D, Na B, Kwon SJ, Lee D, Kang W, Moon I (2022a) Maximum likelihood training of implicit nonlinear diffusion model. In: Advances in neural information processing systems, vol 35, pp 32270–32284

Kim G, Kwon T, Ye JC (2022b) Diffusionclip: Text-guided diffusion models for robust image manipulation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 2426–2435

Kim D, Lai C-H, Liao W-H, Murata N, Takida Y, Uesaka T, He Y, Mitsufuji Y, Ermon S (2023) Consistency trajectory models: Learning probability flow ode trajectory of diffusion. Preprint. arXiv:2310.02279

Kingma DP, Dhariwal P (2018) Glow: generative flow with invertible 1 × 1 convolutions. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, 10236–10245

Kingma DP, Welling M (2013) Auto-encoding variational Bayes. Preprint. arXiv:1312.6114

Kingma D, Salimans T, Poole B, Ho J (2021) Variational diffusion models. In: Advances in neural information processing systems, vol 34, pp 21696–21707

Kong Z, Ping W (2021) On fast sampling of diffusion probabilistic models. Preprint. arXiv:2106.00132

Krizhevsky A, Hinton G et al (2009) Learning multiple layers of features from tiny images. Technical Report

Kulikov V, Yadin S, Kleiner M, Michaeli T (2023) SinDDM: a single image denoising diffusion model. In: International conference on machine learning, pp 17920–17930. PMLR

Kumar A, Raghunathan A, Jones R, Ma T, Liang P (2022) Fine-tuning can distort pretrained features and underperform out-of-distribution. In: International conference on learning representations

Kumari N, Zhang B, Zhang R, Shechtman E, Zhu J-Y (2023) Multi-concept customization of text-to-image diffusion. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1931–1941

Kynkäänniemi T, Karras T, Laine S, Lehtinen J, Aila T (2019) Improved precision and recall metric for assessing generative models. In: Advances in neural information processing systems, vol 32

Laousy O, Araujo A, Chassagnon G, Revel M-P, Garg S, Khorrami F, Vakalopoulou M (2023) Towards better certified segmentation via diffusion models. Preprint. arXiv:2306.09949

Lee Y, Kim J-Y, Go H, Jeong M, Oh S, Choi S (2023a) Multi-architecture multi-expert diffusion models. Preprint. arXiv:2306.04990

Lee S, Kim B, Ye JC (2023b) Minimizing trajectory curvature of ODE-based generative models. Preprint. arXiv:2301.12003

Lee K, Liu H, Ryu M, Watkins O, Du Y, Boutilier C, Abbeel P, Ghavamzadeh M, Gu SS (2023c) Aligning text-to-image models using human feedback. Preprint. arXiv:2302.12192

Lee Y, Park K, Cho Y, Lee Y-J, Hwang SJ (2023d) KOALA: self-attention matters in knowledge distillation of latent diffusion models for memory-efficient and fast image synthesis. Preprint. arXiv:2312.04005

Lemercier J-M, Richter J, Welker S, Gerkmann T (2023) StoRM: a diffusion-based stochastic regeneration model for speech enhancement and dereverberation. IEEE/ACM Trans Audio Speech Lang Process 31:2724–2737

Leng Y, Huang Q, Wang Z, Liu Y, Zhang H (2023) DiffuseGAE: controllable and high-fidelity image manipulation from disentangled representation. Preprint. arXiv:2307.05899

Li X, Thickstun J, Gulrajani I, Liang PS, Hashimoto TB (2022a) Diffusion-LM improves controllable text generation. In: Advances in neural information processing systems, vol 35, pp 4328–4343

Li M, Lin J, Meng C, Ermon S, Han S, Zhu J-Y (2022b) Efficient spatially sparse inference for conditional GANs and diffusion models. In: Advances in neural information processing systems, vol 35, pp 28858–28873

Li H, Yang Y, Chang M, Chen S, Feng H, Xu Z, Li Q, Chen Y (2022c) SRDiff: single image super-resolution with diffusion probabilistic models. Neurocomputing 479:47–59

Li W, Yu X, Zhou K, Song Y, Lin Z, Jia J (2022d) Image Inpainting via Iteratively decoupled probabilistic modeling. Preprint. arXiv:2212.02963

Li X, Lian L, Liu Y, Yang H, Dong Z, Kang D, Zhang S, Keutzer K (2023a) Q-Diffusion: quantizing diffusion models. Preprint. arXiv:2302.04304

Li Y, Wang H, Jin Q, Hu J, Chemerys P, Fu Y, Wang Y, Tulyakov S, Ren J (2023b) SnapFusion: text-to-image diffusion model on mobile devices within two seconds. Preprint. arXiv:2306.00980

Liang J, Zeng H, Zhang L (2022) Efficient and degradation-adaptive network for real-world image super-resolution. In: European conference on computer vision. Springer, Cham, pp 574–591

Lin S, Liu B, Li J, Yang X (2024) Common diffusion noise schedules and sample steps are flawed. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision, pp 5404–5411

Liu X, Ji K, Fu Y, Tam W, Du Z, Yang Z, Tang J (2022) P-Tuning: prompt tuning can be comparable to fine-tuning across scales and tasks. In: Proceedings of the 60th annual meeting of the association for computational linguistics, vol 2: short papers, pp 61–68

Liu Z, Guo Y, Yu K (2023a) Diffvoice: Text-to-speech with latent diffusion. In: ICASSP 2023-2023 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 1–5

Liu E, Ning X, Lin Z, Yang H, Wang Y (2023b) OMS-DPM: optimizing the model schedule for diffusion probabilistic models. Preprint. arXiv:2306.08860

Lu C, Zhou Y, Bao F, Chen J, Li C, Zhu J (2022a) DPM-Solver: a fast ODE solver for diffusion probabilistic model sampling in around 10 steps. In: Advances in neural information processing systems, vol 35, pp 5775–5787

Lu C, Zhou Y, Bao F, Chen J, Li C, Zhu J (2022b) DPM-Solver++: fast solver for guided sampling of diffusion probabilistic models. Preprint. arXiv:2211.01095

Lu S, Liu Y, Kong AW-K (2023) TF-ICON: diffusion-based training-free cross-domain image composition. Preprint. arXiv:2307.12493

Luhman E, Luhman T (2021) Knowledge distillation in iterative generative models for improved sampling speed. Preprint. arXiv:2101.02388

Luo F, Xiang J, Zhang J, Han X, Yang W (2023a) Image super-resolution via latent diffusion: a sampling-space mixture of experts and frequency-augmented decoder approach. Preprint. arXiv:2310.12004

Luo S, Tan Y, Huang L, Li J, Zhao H (2023b) Latent consistency models: synthesizing high-resolution images with few-step inference. Preprint. arXiv:2310.04378

Luo S, Tan Y, Patil S, Gu D, Platen P, Passos A, Huang L, Li J, Zhao H (2023c) LCM-LoRA: a universal stable-diffusion acceleration module. Preprint. arXiv:2311.05556

Ma J, Hu T, Wang W, Sun J (2023a) Elucidating the design space of classifier-guided diffusion generation. Preprint. arXiv:2310.11311

Ma Z, Li J, Zhou B et al (2023b) LMD: faster image reconstruction with latent masking diffusion. Preprint. arXiv:2312.07971

Macha S, Oza O, Escott A, Caliva F, Armitano R, Cheekatmalla SK, Parthasarathi SHK, Liu Y (2023) Fixed-point quantization aware training for on-device keyword-spotting. In: ICASSP 2023-2023 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 1–5

Mahajan D, Girshick R, Ramanathan V, He K, Paluri M, Li Y, Bharambe A, Van Der Maaten L (2018) Exploring the limits of weakly supervised pretraining. In: Proceedings of the European conference on computer vision (ECCV), pp 181–196

Manukyan H, Sargsyan A, Atanyan B, Wang Z, Navasardyan S, Shi H (2023) HD-Painter: high-resolution and prompt-faithful text-guided image inpainting with diffusion models. Preprint. arXiv:2312.14091

Mao W, Xu C, Zhu Q, Chen S, Wang Y (2023) Leapfrog diffusion model for stochastic trajectory prediction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 5517–5526

Masip S, Rodriguez P, Tuytelaars T, Ven GM (2023) Continual learning of diffusion models with generative distillation. Preprint. arXiv:2311.14028

Mei K, Patel V (2023) VIDM: video implicit diffusion models. In: Proceedings of the AAAI conference on artificial intelligence, vol 37, pp 9117–9125

Mei K, Delbracio M, Talebi H, Tu Z, Patel VM, Milanfar P (2023) Conditional diffusion distillation. Preprint. arXiv:2310.01407

Meng C, He Y, Song Y, Song J, Wu J, Zhu J-Y, Ermon S (2021) SDEdit: guided image synthesis and editing with stochastic differential equations. In: International conference on learning representations

Meng C, Rombach R, Gao R, Kingma D, Ermon S, Ho J, Salimans T (2023) On distillation of guided diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 14297–14306

Moser B.B, Frolov S, Raue F, Palacio S, Dengel A (2023) Waving goodbye to low-res: a diffusion-wavelet approach for image super-resolution. CoRR. arXiv:2304.01994

Nash C, Menick J, Dieleman S, Battaglia P (2021) Generating images with sparse representations. In: International conference on machine learning, pp 7958–7968. PMLR

Nguyen TH, Tran A (2023) SwiftBrush: one-step text-to-image diffusion model with variational score distillation. Preprint. arXiv:2312.05239

Ni H, Shi C, Li K, Huang SX, Min MR (2023) Conditional image-to-video generation with latent flow diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 18444–18455

Nichol AQ, Dhariwal P (2021) Improved denoising diffusion probabilistic models. In: International conference on machine learning, pp 8162–8171. PMLR

Nichol A.Q, Dhariwal P, Ramesh A, Shyam P, Mishkin P, Mcgrew B, Sutskever I, Chen M (2022) Glide: towards photorealistic image generation and editing with text-guided diffusion models. In: International conference on machine learning, pp 16784–16804. PMLR

Nie S, Guo HA, Lu C, Zhou Y, Zheng C, Li C (2023) The blessing of randomness: SDE beats ODE in general diffusion-based image editing. Preprint. arXiv:2311.01410

Ning M, Sangineto E, Porrello A, Calderara S, Cucchiara R (2023) Input perturbation reduces exposure bias in diffusion models. Preprint. arXiv:2301.11706

Niu A, Trung PX, Zhang K, Sun J, Zhu Y, Kweon IS, Zhang Y (2023) ACDMSR: accelerated conditional diffusion models for single image super-resolution. Preprint. arXiv:2307.00781

Oh S, Sim H, Kim J, Lee J (2022) Non-uniform step size quantization for accurate post-training quantization. In: European conference on computer vision. Springer, Cham, pp 658–673

Okamoto T, Toda T, Shiga Y, Kawai H (2021) Noise level limited sub-modeling for diffusion probabilistic vocoders. In: ICASSP 2021-2021 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 6029–6033

Pandey K, Mukherjee A, Rai P, Kumar A (2022) DiffuseVAE: efficient, controllable and high-fidelity generation from low-dimensional latents. Preprint. arXiv:2201.00308

Pandey K, Rudolph M, Mandt S (2023) Efficient integrators for diffusion generative models. Preprint. arXiv:2310.07894

Peebles W, Xie S (2022) Scalable diffusion models with transformers. Preprint. arXiv:2212.09748

Permenter F, Yuan C (2023) Interpreting and improving diffusion models using the Euclidean distance function. Preprint. arXiv:2306.04848

Phung H, Dao Q, Tran A (2023) Wavelet diffusion models are fast and scalable image generators. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10199–10208

Popov V, Vovk I, Gogoryan V, Sadekova T, Kudinov M (2021) Grad-TTS: a diffusion probabilistic model for text-to-speech. In: International conference on machine learning, pp 8599–8608. PMLR

Preechakul K, Chatthee N, Wizadwongsa S, Suwajanakorn S (2022) Diffusion autoencoders: toward a meaningful and decodable representation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10619–10629

Qian L, Wang M, Liu Y, Zhou H (2022) Diff-Glat: diffusion glancing transformer for parallel sequence to sequence learning. Preprint. arXiv:2212.10240

Ramesh A, Dhariwal P, Nichol A, Chu C, Chen M (2022) Hierarchical text-conditional image generation with clip latents. Preprint. arXiv:2204.06125

Rombach R, Blattmann A, Lorenz D, Esser P, Ommer B (2022) High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10684–10695

Ruan L, Ma Y, Yang H, He H, Liu B, Fu J, Yuan NJ, Jin Q, Guo B (2023) MM-Diffusion: learning multimodal diffusion models for joint audio and video generation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10219–10228

Ruiz N, Li Y, Jampani V, Pritch Y, Rubinstein M, Aberman K (2023a) DreamBooth: fine tuning text-to-image diffusion models for subject-driven generation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 22500–22510

Ruiz N, Li Y, Jampani V, Wei W, Hou T, Pritch Y, Wadhwa N, Rubinstein M, Aberman K (2023b) HyperDreamBooth: hypernetworks for fast personalization of text-to-image models. Preprint. arXiv:2307.06949

Ryu D, Ye JC (2022) Pyramidal denoising diffusion probabilistic models. Preprint. arXiv:2208.01864

Saharia C, Chan W, Saxena S, Li L, Whang J, Denton EL, Ghasemipour K, Gontijo Lopes R, Karagol Ayan B, Salimans T et al (2022a) Photorealistic text-to-image diffusion models with deep language understanding. In: Advances in neural information processing systems, vol 35, pp 36479–36494

Saharia C, Ho J, Chan W, Salimans T, Fleet DJ, Norouzi M (2022b) Image super-resolution via iterative refinement. IEEE Trans Pattern Anal Mach Intell 45(4):4713–4726

Salimans T, Ho J (2021) Progressive distillation for fast sampling of diffusion models. In: International conference on learning representations

Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A, Chen X (2016) Improved techniques for training GANs. In: Advances in neural information processing systems, vol 29

Sanh V, Wolf T, Rush A (2020) Movement pruning: adaptive sparsity by fine-tuningg. In: Advances in neural information processing systems, vol 33, pp 20378–20389

Sauer A, Lorenz D, Blattmann A, Rombach R (2023) Adversarial diffusion distillation. Preprint. arXiv:2311.17042

Scarvelis C, Borde HSdO, Solomon J (2023) Closed-form diffusion models. Preprint. arXiv:2310.12395

Sekhar Sahoo S, Gokaslan A, De Sa C, Kuleshov V (2023) Diffusion models with learned adaptive noise. Preprint. arXiv:2312.13236

Shang S, Shan Z, Liu G, Zhang J (2023a) ResDiff: combining cnn and diffusion model for image super-resolution. Preprint. arXiv:2303.08714

Shang Y, Yuan Z, Xie B, Wu B, Yan Y (2023b) Post-training quantization on diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1972–1981

Shao S, Dai X, Yin S, Li L, Chen H, Hu Y (2023) Catch-up distillation: you only need to train once for accelerating sampling. Preprint. arXiv:2305.10769

Shen L, Yan J, Sun X, Li B, Pan Z (2023) Wavelet-based self-attention GAN with collaborative feature fusion for image inpainting. IEEE Trans Emerg Top Comput Intell 7:1651–1664

So J, Lee J, Ahn D, Kim H, Park E (2023) Temporal dynamic quantization for diffusion models. Preprint. arXiv:2306.02316

Song Y, Dhariwal P (2023) Improved techniques for training consistency models. Preprint. arXiv:2310.14189

Song J, Meng C, Ermon S (2020a) Denoising diffusion implicit models. In: International conference on learning representations

Song Y, Sohl-Dickstein J, Kingma DP, Kumar A, Ermon S, Poole B (2020b) Score-based generative modeling through stochastic differential equations. In: International conference on learning representations

Song W, Ma W, Ma Y, Zhao X, Lin G (2022) Improving the spatial resolution of solar images based on an improved conditional denoising diffusion probability model. Astrophys J Suppl Ser 263(2):25

Song Y, Dhariwal P, Chen M, Sutskever I (2023) Consistency models. Preprint. arXiv:2303.01469

Starodubcev N, Fedorov A, Babenko A, Baranchuk D (2023) Your student is better than expected: adaptive teacher–student collaboration for text-conditional diffusion models. Preprint. arXiv:2312.10835

Strang G (1968) On the construction and comparison of difference schemes. SIAM J Numer Anal 5(3):506–517

Sun W, Chen D, Wang C, Ye D, Feng Y, Chen C (2022) Accelerating diffusion sampling with classifier-based feature distillation. Preprint. arXiv:2211.12039

Tang T, Chen Y, Du Y, Li J, Zhao WX, Wen J-R (2023) Learning to Imagine: visually-augmented natural language generation. Preprint. arXiv:2305.16944

Tsaban L, Passos A (2023) LEDITS: real image editing with DDPM inversion and semantic guidance. Preprint. arXiv:2307.00522

Uria B, Côté M-A, Gregor K, Murray I, Larochelle H (2016) Neural autoregressive distribution estimation. J Mach Learn Res 17(1):7184–7220

Vahdat A, Kreis K (2021) Kautz J Score-based generative modeling in latent space. In: Advances in neural information processing systems, vol 34, pp 11287–11302

Voronov A, Khoroshikh M, Babenko A, Ryabinin M (2023) Is this loss informative? speeding up textual inversion with deterministic objective evaluation. Preprint. arXiv:2302.04841

Wang X, Yan J-K, Cai J-Y, Deng J-H, Qin Q, Wang Q, Xiao H, Cheng Y, Ye P-F (2022a) Superresolution reconstruction of single image for latent features. Preprint. arXiv:2211.12845

Wang T, Zhang T, Zhang B, Ouyang H, Chen D, Chen Q, Wen F (2022b) Pretraining is all you need for image-to-image translation. Preprint. arXiv:2205.12952

Wang Z, Zheng H, He P, Chen W, Zhou M (2022c) Diffusion-Gan: training GANs with diffusion. In: The 11th International conference on learning representations

Wang Z, Wang J, Liu Z, Qiu Q (2023a) Binary latent diffusion. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 22576–22585

Wang C, Wang Z, Xu X, Tang Y, Zhou J, Lu J (2023b) Towards accurate data-free quantization for diffusion models. Preprint. arXiv:2305.18723

Wang Y, Yang W, Chen X, Wang Y, Guo L, Chau L-P, Liu Z, Qiao Y, Kot AC, Wen B (2023c) SinSR: diffusion-based image super-resolution in a single step. Preprint. arXiv:2311.14760

Watson D, Ho J, Norouzi M, Chan W (2021) Learning to efficiently sample from diffusion probabilistic models. Preprint. arXiv:2106.03802

Wei X, Gong R, Li Y, Liu X, Yu F (2021) Qdrop: Randomly dropping quantization for extremely low-bit post-training quantization. In: International conference on learning representations

Wizadwongsa S, Suwajanakorn S (2022) Accelerating guided diffusion sampling with splitting numerical methods. In: The Eleventh International Conference on Learning Representations

Wortsman M, Ilharco G, Kim JW, Li M, Kornblith S, Roelofs R, Lopes R.G, Hajishirzi H, Farhadi A, Namkoong H et al (2022) Robust fine-tuning of zero-shot models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 7959–7971

Wu Q, Liu Y, Zhao H, Kale A, Bui T, Yu T, Lin Z, Zhang Y, Chang S (2023a) Uncovering the disentanglement capability in text-to-image diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1900–1910

Wu Z, Zhou P, Kawaguchi K, Zhang H (2023b) Fast diffusion model. Preprint. arXiv:2306.06991

Xia W, Cong W, Wang G (2022) Patch-based denoising diffusion probabilistic model for sparse-view CT reconstruction. Preprint. arXiv:2211.10388

Xia M, Shen Y, Lei C, Zhou Y, Yi R, Zhao D, Wang W, Liu Y-j (2023a) Towards more accurate diffusion model acceleration with a timestep aligner. Preprint. arXiv:2310.09469

Xia B, Zhang Y, Wang S, Wang Y, Wu X, Tian Y, Yang W, Van Gool L (2023b) DiffIR: efficient diffusion model for image restoration. Preprint. arXiv:2303.09472

Xiao Z, Kreis K, Vahdat A (2021) Tackling the generative learning trilemma with denoising diffusion gans. In: International conference on learning representations

Xiao J, Yin M, Gong Y, Zang X, Ren J, Yuan B (2023a) COMCAT: towards efficient compression and customization of attention-based vision models. Preprint. arXiv:2305.17235

Xiao Y, Yuan Q, Jiang K, He J, Jin X, Zhang L (2023b) EDiffSR: an efficient diffusion probabilistic model for remote sensing image super-resolution. IEEE Trans Geosci Remot Sens 62:5601514

Xie E, Yao L, Shi H, Liu Z, Zhou D, Liu Z, Li J, Li Z (2023) DiffFit: unlocking transferability of large diffusion models via simple parameter-efficient fine-tuning. Preprint. arXiv:2304.06648

Xu Y, Gong M, Xie S, Wei W, Grundmann M, Hou T et al (2023) Semi-implicit denoising diffusion models (SIDDMs). Preprint. arXiv:2306.12511

Yang S, Chen Y, Wang L, Liu S, Chen Y (2023a) Denoising diffusion step-aware models. Preprint. arXiv:2310.03337

Yang B, Gu S, Zhang B, Zhang T, Chen X, Sun X, Chen D, Wen F (2023b) Paint by example: exemplar-based image editing with diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 18381–18391

Yang X, Zhou D, Feng J, Wang X (2023c) Diffusion probabilistic model made slim. In: Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition, pp 22552–22562

Yin G, Wang W, Yuan Z, Han C, Ji W, Sun S, Wang C (2022) Content-variant reference image quality assessment via knowledge distillation. In: Proceedings of the AAAI conference on artificial intelligence, vol 36, pp 3134–3142

Yin T, Gharbi M, Zhang R, Shechtman E, Durand F, Freeman WT, Park T (2023) One-step diffusion with distribution matching distillation. Preprint. arXiv:2311.18828

Youn J, Song J, Kim H-S, Bahk S (2022) Bitwidth-adaptive quantization-aware neural network training: a meta-learning approach. In: European conference on computer vision. Springer, Cham, pp 208–224

Yu F, Seff A, Zhang Y, Song S, Funkhouser T, Xiao J (2015) LSUN: construction of a large-scale image dataset using deep learning with humans in the loop. Preprint. arXiv:1506.03365

Yue Z, Wang J, Loy CC (2023) ResShift: efficient diffusion model for image super-resolution by residual shifting. Preprint. arXiv:2307.12348

Yu H, Shen L, Huang J, Zhou M, Li H, Zhao F (2023a) Debias the training of diffusion models. Preprint. arXiv:2310.08442

Yu S, Sohn K, Kim S, Shin J (2023b) Video probabilistic diffusion models in projected latent space. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 18456–18466

Zhang Q (2021) Diffusion normalizing flow. In: Advances in neural information processing systems, vol 34, pp 16280–16291

Zhang Q, Chen Y (2022) Fast sampling of diffusion models with exponential integrator. In: The 11th International conference on learning representations

Zhang L, Agrawala M (2023) Adding conditional control to text-to-image diffusion models. Preprint. arXiv:2302.05543

Zhang K, Liang J, Van Gool L, Timofte R (2021) Designing a practical degradation model for deep blind image super-resolution. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 4791–4800

Zhang Z, Zhao Z, Lin Z (2022a) Unsupervised representation learning from pre-trained diffusion probabilistic models. In: Advances in neural information processing systems, vol 35, pp 22117–22130

Zhang L, Chen X, Tu X, Wan P, Xu N, Ma K (2022b) Wavelet knowledge distillation: towards efficient image-to-image translation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 12464–12474

Zhang Z, Ehinger KA, Drummond T (2023a) Improving denoising diffusion models via simultaneous estimation of image and noise. Preprint. arXiv:2310.17167

Zhang H, Feng R, Yang Z, Huang L, Liu Y, Zhang Y, Shen Y, Zhao D, Zhou J, Cheng F (2023b) Dimensionality-varying diffusion process. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 14307–14316

Zhang L, Ma H, Zhu X, Feng J (2023c) Preconditioned score-based generative models. Preprint. arXiv:2302.06504

Zhang H, Wang Z, Wu Z, Jiang Y-G (2023d) DiffusionAD: denoising diffusion for anomaly detection. Preprint. arXiv:2303.08730

Zhang K, Yang X, Wang WY, Li L (2023e) ReDi: efficient learning-free diffusion inference via trajectory retrieval. Preprint. arXiv:2302.02285

Zhao W, Bai L, Rao Y, Zhou J, Lu J (2023a) UniPC: a unified predictor-corrector framework for fast sampling of diffusion models. Preprint. arXiv:2302.04867

Zhao K, Hung ALY, Pang K, Zheng H, Sung K (2023b) PartDiff: image super-resolution with partial diffusion models. Preprint. arXiv:2307.11926

Zhao C, Yang P, Zhou F, Yue G, Wang S, Wu H, Chen G, Wang T, Lei B (2023c) MHW-GAN: multidiscriminator hierarchical wavelet generative adversarial network for multimodal image fusion. IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2023.3271059

Zheng H, Nie W, Vahdat A, Anandkumar A (2023a) Fast training of diffusion models with masked transformers. Preprint. arXiv:2306.09305

Zheng H, Nie W, Vahdat A, Azizzadenesheli K, Anandkumar A (2023b) Fast sampling of diffusion models via operator learning. In: International conference on machine learning, pp 42390–42402. PMLR

Zheng H, Wang Z, Yuan J, Ning G, He P, You Q, Yang H, Zhou M (2023c) Learning stackable and skippable LEGO bricks for efficient, reconfigurable, and variable-resolution diffusion modeling. Preprint. arXiv:2310.06389

Zhou Z, Chen D, Wang C, Chen C (2023a) Fast ODE-based sampling for diffusion models in around 5 steps. Preprint. arXiv:2312.00094

Zhou D, Yang Z, Yang Y (2023b) Pyramid diffusion models for low-light image enhancement. Preprint. arXiv:2305.10028

Zhu J, Ma H, Chen J, Yuan J (2023) DomainStudio: fine-tuning diffusion models for domain-driven image generation using limited data. Preprint. arXiv:2306.14153