

Conditional Image Synthesis with Diffusion Models: A Survey

Zheyuan Zhan, Defang Chen, Jian-Ping Mei, Zhenghe Zhao, Jiawei Chen, Chun Chen,
Siwei Lyu, *Fellow, IEEE* and Can Wang

Abstract—Conditional image synthesis based on user-specified requirements is a key component in creating complex visual content. In recent years, diffusion-based generative modeling has become a highly effective way for conditional image synthesis, leading to exponential growth in the literature. However, the complexity of diffusion-based modeling, the wide range of image synthesis tasks, and the diversity of conditioning mechanisms present significant challenges for researchers to keep up with rapid developments and understand the core concepts on this topic. In this survey, we categorize existing works based on how conditions are integrated into the two fundamental components of diffusion-based modeling, *i.e.*, the denoising network and the sampling process. We specifically highlight the underlying principles, advantages, and potential challenges of various conditioning approaches in the training, re-purposing, and specialization stages to construct a desired denoising network. We also summarize six mainstream conditioning mechanisms in the essential sampling process. All discussions are centered around popular applications. Finally, we pinpoint some critical yet still open problems to be solved in the future and suggest some possible solutions. Our reviewed works are itemized at <https://github.com/zju-pi/Awesome-Conditional-Diffusion-Models>.

Index Terms—Generative Models, Diffusion Models, Conditional Image Synthesis, Condition Integration.

I. INTRODUCTION

Image synthesis is an essential generative AI task. It is more useful when incorporating user-provided conditions to generate images that meet diverse user needs through precise control. Early works have made significant breakthroughs in various conditional image synthesis tasks, such as text-to-image generation [1–5], image restoration [6–9], and image editing [10–12]. However, the performance of conditional image synthesis with early deep learning-based generative models such as generative adversarial networks (GANs) [13, 14], variational auto-encoders (VAEs) [15, 16], and auto-regressive models (ARMs) [17, 18] is unsatisfactory due to their intrinsic limitations: GANs are vulnerable to

Zheyuan Zhan is with the College of Computer Science, Zhejiang University, Hangzhou 310027, China. E-mail: zhanzheyuan@zju.edu.cn.

Defang Chen is with the Department of Computer Science and Engineering, University at Buffalo, State University of New York, Buffalo, NY 14260, USA. E-mail: defchern@gmail.com.

Jian-Ping Mei is with the College of Computer Science, Zhejiang University of Technology, Hangzhou 310027, China. E-mail: jpmei@zjut.edu.cn

Zhenghe Zhao, Jiawei Chen, Can Wang and Chun Chen are with the College of Computer Science, Zhejiang University, Hangzhou 310027, China. E-mail: {zhaozhenghe,sleepyhunt,wcan,chencc}@zju.edu.cn.

Siwei Lyu is with the Department of Computer Science and Engineering, University at Buffalo, State University of New York, Buffalo, NY 14260, USA. E-mail: siweilyu@buffalo.edu.

(Corresponding author: Defang Chen)

mode collapse and unstable training [13]; VAEs often generate blurry images [15]; and ARMs suffer from sequential error accumulation and huge time consumption [17].

In recent years, diffusion models (DMs) have emerged as state-of-the-art image generation models due to their strong generative capabilities and versatility [167, 176–179]. In DMs, images are synthesized from Gaussian noise through iterative denoising steps guided by the predictions of a denoising network. This distinctive multi-step sampling process enables DMs to achieve remarkable generative performance characterized by stable training, diverse outputs, and exceptional sample quality. It also gives DMs a unique advantage in facilitating conditional integration compared to one-step generative models. These benefits have made DMs the tool of choice for conditional image synthesis, leading to rapid growth in the research on *Diffusion-based Conditional Image Synthesis* (DCIS) over the past few years [19, 20, 25, 45, 70, 88, 89, 117, 120, 123, 169].

The rapidly expanding body of works, the numerous variations in model architectures, training methods, and sampling techniques, along with the broad scope of potential conditional synthesis tasks, make it challenging for researchers to grasp the full landscape of DCIS. This complexity can be particularly overwhelming for newcomers to the field. What is needed is a systematic survey that offers a comprehensive yet structured overview of this growing research area.

There exist several surveys on specific conditional image synthesis tasks, such as image restoration [180], text-to-image synthesis [181], and image editing [182], or classifying works in computer vision according to their target conditional synthesis tasks [183, 184]. While these task-oriented surveys provide valuable insights into approaches for their respective target tasks, they do not include the commonalities in model frameworks across different conditional synthesis tasks in terms of model architectures and conditioning mechanisms. Two recent surveys [185, 186] provide overview on DM-based works for a wide range of tasks in the field of conditional image synthesis. However, their scope remains limited as they primarily focus on DCIS works built on T2I backbones, neglecting earlier works that integrate conditioning into unconditional denoising networks or involve training task-specific conditional denoising networks from scratch. These earlier efforts are foundational for the current advancements in DCIS using T2I backbones and are still widely applied in low-level tasks such as image restoration. Besides, [185] focuses most of its attention on the DM-based image editing framework and lacks systematic analysis on the unified framework for other

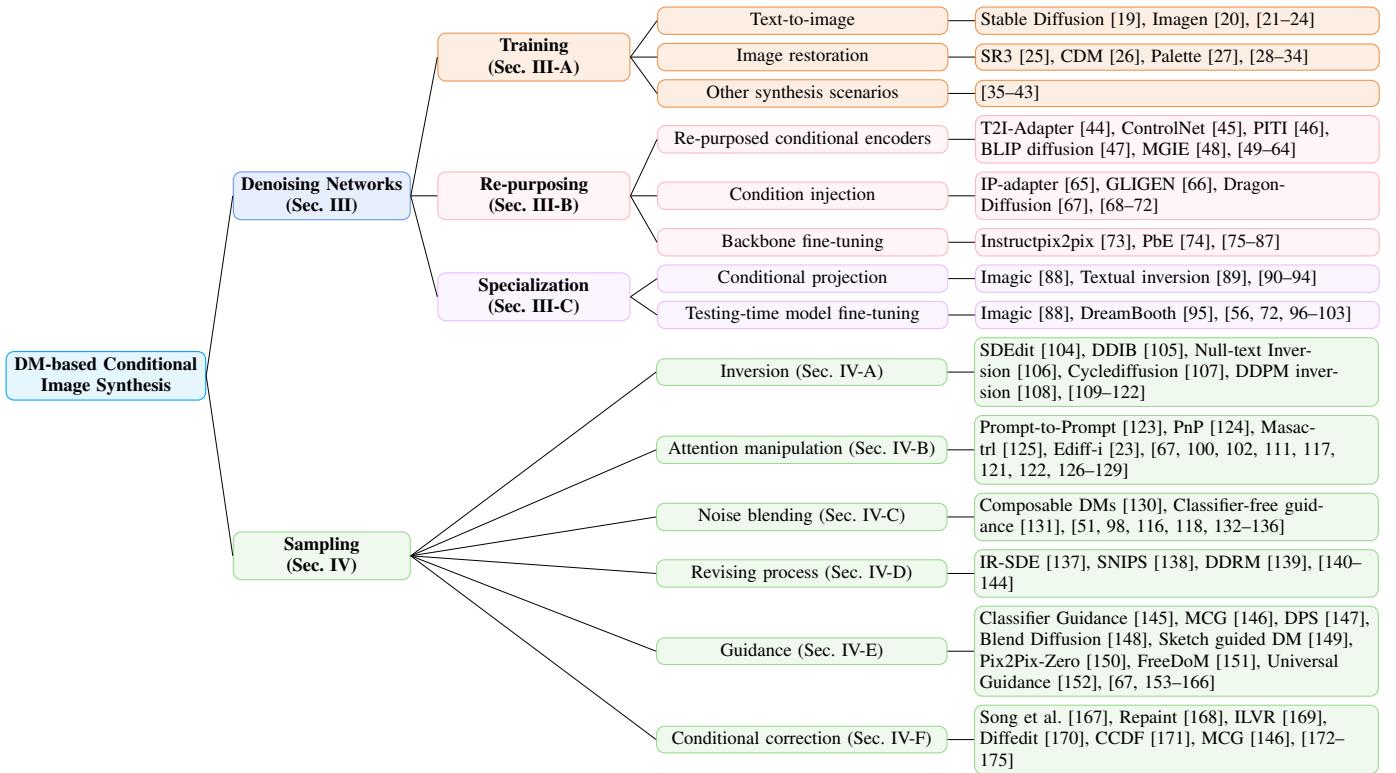


Fig. 1: The proposed taxonomy of diffusion-based conditional image synthesis in this survey. See texts for details.

tasks in this field while [186] does not delve deeper into the design choices in model architecture and detailed conditioning mechanisms for sampling process. This leads to a lack of systematization in their taxonomies and the omission of crucial related works in the field of DCIS.

In contrast, this survey aims to provide a comprehensive and structured framework that covers a wide range of current DCIS works by offering a taxonomy based on the mainstream techniques for condition integration in DCIS frameworks. We present a clear and systematic breakdown of the components and design choices involved in constructing a DCIS framework with condition integration. Specifically, we review and summarize existing DCIS methods by examining how conditions are integrated into the two fundamental components of diffusion modeling: the *denoising network* and the *sampling process*. For the denoising network, we break down the process of establishing a conditional denoising network into three stages. For the sampling process, we categorize six mainstream in-sampling conditioning mechanisms, detailing how control signals are integrated into various components of the sampling process. The objective is to give readers a high-level and accessible overview of existing DCIS works across diverse tasks, equipping them to design conditional synthesis frameworks for their own desired tasks, including novel tasks that have yet to be explored.

The remainder of this survey is organized as follows: we first introduce the background of diffusion models and the conditional image synthesis task in Sec. II. Next, we summarize methods for condition integration within the denoising network in Sec. III, and for the sampling process in Sec. IV. Finally, we explore potential future directions in Sec. V. Fig. 1

illustrates the DCIS taxonomy proposed in this survey.

II. BACKGROUNDS

Diffusion-based generative modeling adopts a forward diffusion process of gradually adding noise into clean data and learns a denoising network to predict the added noise. In the sampling process, data is synthesized by reversing the forward process from Gaussian noise based on the prediction of a denoising network. We first introduce the core concepts of discrete-time and continuous-time diffusion modeling in Sec. II-A. Then, we discuss the model architecture in Sec. II-B and highlight representative DCIS tasks in Sec. II-C.

A. The Formulation of Diffusion Modeling

1) *Discrete-Time Formulation*: The discrete-time diffusion model was initially proposed in [176]. It constructs a forward Markov chain to transform clean data into noise by progressively adding small amounts of Gaussian noise so that a parameterized denoising network can be learned to predict the added noise in each forward step. Once the denoising network is trained, images can be generated from Gaussian noise by reversing the diffusion process. This idea gained popularity through an important follow-up work known as denoising diffusion probabilistic models (DDPMs) [177]. This work led to a substantial improvement in the quality of synthesized images and increased resolutions, from 32×32 [176] to 256×256 , sparking a rapidly growing interest in diffusion models. Next, we adopt the notation from DDPM [177], which is widely used in the literature to describe discrete-time diffusion models [19, 88, 187].

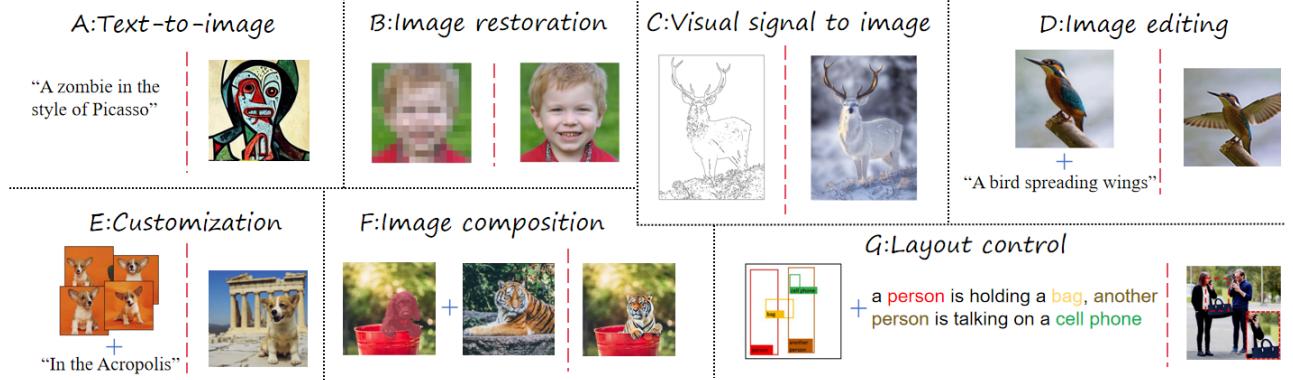


Fig. 2: Seven representative conditional image synthesis tasks with their input/output. Figures are cited from the following papers: (A) Stable Diffusion [19]; (B) SR3 [25]; (C) ControlNet [45]; (D) Imagic [88]; (E) DreamBooth [95]; (F) PbE [74]; (G) InteractDiffusion [69].

The forward Markov chain is parameterized based on a pre-defined schedule β_1, \dots, β_T , where β_t is the noise variance in each step and the total number of steps T is usually large, e.g., 1,000. Given the clean data sampled from the training dataset $\mathbf{x}_0 \sim p_{\text{data}}(\mathbf{x})$, the transition kernel is $q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$, or, $q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$, where $\mathbf{x}_1, \dots, \mathbf{x}_T$ are latent variables, $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$, and $\bar{\alpha}_T \rightarrow 0$. By progressively adding Gaussian noise to the clean data, this Markov chain transforms the data distribution to an approximate normal distribution, i.e., $\int q(\mathbf{x}_T | \mathbf{x}_0)p_{\text{data}}(\mathbf{x}_0)d\mathbf{x}_0 \approx \mathcal{N}(\mathbf{0}, \mathbf{I})$.

In the training phase, DDPM [177] learns a denoising network with parameter θ by minimizing the KL divergence between the transition kernel $p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)$ and the posterior distribution $q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0)$. In practice, DDPM [177] is trained on the following re-parameterized loss function to improve the training stability and sample quality:

$$\mathbb{E}_{t, \mathbf{x}_0, \epsilon} \left[\|\epsilon - \epsilon_\theta (\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\|^2 \right], \quad (1)$$

where $\epsilon_\theta(\mathbf{x}_t, t)$ is a noise-prediction network to estimate the added noise $\epsilon = \frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}}$ in each step. For the conditional generation that performs denoising steps conditioned on control signal \mathbf{c} , the conditional denoising network $\epsilon_\theta(\mathbf{x}_t, t, \mathbf{c})$ can be trained on a loss function similar to Eq. 1.

In the sampling process, DDPM gradually generates clean data from Gaussian noise by computing the reverse transition kernel p_θ with the learned network ϵ_θ , i.e.,

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta \right) + \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \beta_t \epsilon_t, \quad (2)$$

where $\epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the standard Gaussian noise independent of \mathbf{x}_t . The following work DDIM [187] proposed a family of sampling processes sharing the same marginal distribution $p(\mathbf{x}_t)$ with the above sampling process, which are written as

$$\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \cdot \mathbf{f}_\theta(\mathbf{x}_t) + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2} \cdot \epsilon_\theta + \sigma_t \epsilon_t, \quad (3)$$

where $\mathbf{f}_\theta(\mathbf{x}_t) = \frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta}{\sqrt{\bar{\alpha}_t}}$ denotes the predicted \mathbf{x}_0 at time step t . For simplicity, we will refer to $\mathbf{f}_\theta(\mathbf{x}_t)$ as the intermediate denoising output $\mathbf{x}_{0|t}$ hereafter. Each choice of

σ_t represents a specific sampling process in DDIM [187]. It is identical to the DDPM generative process in Eq. 2 when $\sigma_t = \sqrt{(1-\bar{\alpha}_{t-1}) / (1-\bar{\alpha}_t)} \sqrt{1-\bar{\alpha}_t/\bar{\alpha}_{t-1}}$ and becomes a deterministic process when $\sigma_t = 0$.

2) *Continuous-Time Formulation*: Song et al [167] proposed to formulate a diffusion process $\{\mathbf{x}_t \sim p_t(\mathbf{x})\}_{t=0}^T$ with the continuous time variable $t \in [0, T]$ as the solution of an Itô stochastic differential equation (SDE) $d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + g(t)d\mathbf{w}_t$, where \mathbf{w}_t denotes the standard Wiener process, and $\mathbf{f}(\mathbf{x}, t)$ and $g(t)$ are drift and diffusion coefficients, respectively [188, 189]. This diffusion process smoothly transforms a data distribution into an approximate noise distribution p_n and its specific discretization recovers the forward process of DDPM [177]. There exists a probability flow ordinary differential equation (PF-ODE) $d\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - \frac{1}{2}g(t)^2\nabla_{\mathbf{x}} \log p_t(\mathbf{x})] dt$, sharing the same marginal distribution with the reverse SDE $d\mathbf{x} = [\mathbf{f}(\mathbf{x}, t) - g(t)^2\nabla_{\mathbf{x}} \log p_t(\mathbf{x})] dt + g(t)d\hat{\mathbf{w}}$ [167, 178, 179, 190]. Therefore, we can learn a time-dependent score-based denoising network $\mathbf{s}_\theta(\mathbf{x}_t, t)$ to estimate the score function $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$ with a sum of denoising score matching [191, 192] objectives weighted by $\lambda(t)$:

$$\mathbb{E}_t \left[\lambda(t) \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_t} \left[\|\mathbf{s}_\theta(\mathbf{x}_t, t) - \nabla_{\mathbf{x}} \log p(\mathbf{x}_t | \mathbf{x}_0)\|_2^2 \right] \right]. \quad (4)$$

When the score-based denoising network $\mathbf{s}_\theta(\mathbf{x}_t, t)$ is trained, we can employ general-purpose numerical methods such as Euler-Maruyama and Runge-Kutta methods to solve the reverse SDE or PF-ODE and recover clean data \mathbf{x}_0 from \mathbf{x}_T .

In the following sections, unless otherwise specified, we will use notation ϵ_θ to represent the denoising network.

B. Architecture of the Denoising Network

Pioneering works adopted U-Net [193] as the denoising network architecture [177, 187, 194, 195]. A U-Net typically consists of an U-shaped encoder-decoder structure with skip connections. The encoder leverages a stack of residual layers and downsampling convolutions to reduce the spatial data dimension and the decoder upsamples the compressed data back to the original dimension. The U-Net architecture is

advantageous for diffusion models due to its exceptional feature extraction, contextual understanding, precise segmentation, and dimensionality preservation property, which enables accurate modeling of complex data distributions for high-quality synthesis. Many followed-up works improved the vanilla U-Net architecture by incorporating multi-head attention [145, 167, 196], normalization [145, 177, 196], or cross-attention layers[19, 20]. Recently, transformers emerged as an alternative for denoising networks because of its capability in capturing long-range dependencies [197, 198], and have achieved success in DM-based works for many tasks including class-conditional generation [199], text-to-image generation [24, 198, 200–202], layout generation [203], and medical image generation [204]. In the following sections, unless otherwise specified, we assume the architecture of the denoising network adopts a U-Net structure.

C. Conditional Image Synthesis Tasks

A conditional image synthesis task \mathcal{T} generates target image \mathbf{x} by sampling from a conditional distribution:

$$\mathbf{x} \sim p_{\mathcal{T}}(\mathbf{x}|\mathbf{c}), \mathbf{c} \in \mathcal{D}_{\mathcal{T}}, \quad (5)$$

where $\mathcal{D}_{\mathcal{T}}$ is the domain of conditional input \mathbf{c} , and $p_{\mathcal{T}}$ is the conditional distribution defined by the task \mathcal{T} . Based on the form of conditional inputs and the correlation between the conditional input and the target image formulated as conditional distribution $p_{\mathcal{T}}(\mathbf{x}|\mathbf{c})$, we classify representative conditional image synthesis tasks into seven categories as shown in Fig. 2: (a) *Text-to-image* synthesizes images in accordance with text prompts, (b) *Image restoration* recovers clean images from their degraded counterparts, (c) *visual signal to image* converts given visual signals such as sketch, depth and human pose into corresponding images, (d) *Image editing* edits the given source images with provided semantic, structure or style information, (e) *Customization* creates different editing renditions for personal object specified by given images, (f) *Image composition* composes the objects and background specified in different images into a single image, and (g) *Layout control* controls the layout grounding of synthesized images with provided spatial information of foreground objects and background. We have sorted out the associations between various conditional synthesis tasks and conditioning mechanisms of representative existing works in Tab. I.

III. CONDITION INTEGRATION IN DENOISING NETWORKS

The denoising network is the crucial component in the diffusion model (DM)-based synthesis framework, which estimates the noise added in each forward step to reverse the initial Gaussian noise distribution back into the data distribution. In practice, the most straightforward way to achieve conditional control in DM-based synthesis framework is incorporating the conditional inputs into the denoising network. In this section, we divide the condition integration in denoising network into three stages: (a) *training stage*: training a denoising network on paired conditional input and target image from scratch, (b) *re-purposing stage*: re-purposing a pre-trained denoising network to conditional synthesis scenarios beyond the task it

was trained on, (c) *specialization stage*: Performing testing-time adjustments on denoising network based on user-specified conditional input. Fig. 3 provides an exemplar workflow to build desired denoising network for conditional synthesis tasks including text-to-image, visual signals to image and customization via these three condition integration stages. Next, we first review the fundamental conditional DMs modeled in *training stage* in Sec. III-A. We then summarize the architecture design choices and condition injection approaches in *re-purposing stage* in Sec. III-B. Finally, we introduce the works performing condition integration in *specialization stage* in Sec. III-C.

A. Condition Integration in the Training Stage

The most straightforward way to integrate the conditional control signal \mathbf{c} into the denoising network is performing supervised training from scratch with the following loss function:

$$\mathbb{E}_{\mathbf{c}, \mathbf{x} \sim p(\mathbf{x}|\mathbf{c}), \epsilon, t} \left[\|\epsilon - \epsilon_{\theta}(\mathbf{x}_t, t, \mathbf{c})\|_2^2 \right], \quad (6)$$

where \mathbf{c} and \mathbf{x} denote the paired conditional inputs and target image. Thereby, the learned conditional denoising network $\epsilon_{\theta}(\mathbf{x}_t, t, \mathbf{c})$ can be employed to sample from $p(\mathbf{x}|\mathbf{c})$.

Next, we introduce the existing conditional denoising networks trained from scratch, focusing their model architectures, conditioning mechanisms, which are crucial for creating the connection between the conditional inputs and its corresponding image. Because of the conditioning architectures and mechanisms are designed based on the target scenarios, we categorize these works based on the their applications, represented by text-to-image and image restoration.

1) *Conditional Models for Text-to-Image (T2I)*: Text-to-image is a fundamental task in the field of conditional image synthesis, which establishes the connection between images and the semantic space of text descriptions. Because of the expressiveness of the text semantic space, text-to-image DMs always serve as the *backbone* for more complicated conditional synthesis tasks including image editing [73, 88, 123], customization [89, 95], visual signal to image [44, 45], image composition [74] and layout control [66, 70].

The main challenge in modeling a effective text-to-image framework lies in (a) precisely capture the users' intention described in text prompts and (b) build the connection between text and image in acceptable computational cost. In practice, DM-based text-to-image works design different text encoders base on Transformer encoder [19, 21], CLIP [22–24] or more powerful large language models [20, 23] to extract the features from user provided text prompts. For computational efficiency, these works often train the DMs on a low-dimension space including compressed latent space [19, 24] and low-resolution pixel space [20–23], and subsequently enlarge the resolution of the synthesized results.

Next, we introduce representative text-to-image model: Stable Diffusion [19] and Imagen [20], which serve as the *T2I backbone* for various conditional synthesis tasks.

Similar to VQ-VAE [218] and VQ-GAN [4], Stable Diffusion [19] employs a pre-trained autoencoder to compress the generative space into a low-dimensional latent space

TABLE I: Stack of conditioning mechanisms of mainstream synthesis tasks applied to denoising network and sampling process, respectively. Conditioning encoder indicates the module to convert conditional inputs into task-related feature embedding, where * indicates that the encoder is determined by the specific restoration task. ♠, ♡, ♣, ♦ denote the four re-purposing stage condition injection methods described in Sec. III-B2.

Stack of conditioning mechanisms for denoising network						
Task	Training (backbone)	Conditional encoder	Condition Injection	Backbone fine-tuning	Specialization	Model
Text-to-image	✓	CLIP, BERT, LLMs	♡	✗	✗	[19–24]
Image restoration	✓	Non.	♠	✗	✗	[25, 26, 28]
	✓	*	♠, ♡	✗	✗	[30–34, 205, 206]
	✗ (SD [19])	LLMs-based	♡	✓	✗	[48, 62–64]
Image editing	✗ (T2I DM [19])	Non.	♠	✓	✗	[73, 75, 77–80, 207]
	✗ (T2I DM [19, 20])	Non./BLIP	♡	✗	✓	[88, 90–94, 97, 98, 166]
Customization	✗ (T2I DM [19])	ViT (CLIP)-based	♡, ♦	✗	Optional	[54–57, 59–61]
	✗ (T2I DM [19, 20])	Non.	♡	✗	✓	[72, 89, 95, 99, 101–103]
Visual to image	✗ (T2I DM [19])	Convolution-based	♣	✗	✗	[44, 45, 50, 51, 208–212]
	✗ (T2I DM [19, 21])	ViT-based	♡	✗	✗	[46, 53]
Image composition	✗ (T2I DM [19])	Convolution-based	♡	✓	✗	[58, 213, 214]
	✗ (T2I DM [19])	ViT (CLIP)-based	♡, ♦	✓	✗	[74, 81–86]
Layout control	✗ (T2I DM [19])	ViT (CLIP)-based	♦	✗	✗	[66, 69, 70]
Stack of conditioning mechanisms for sampling process						
Task	Backbone model		Conditioning mechanism		Model	
Text-to-image	Uncond DM		Guidance		[157, 215]	
	Conditional restoration DM [137, 141]		Revising Diffusion Process		[137, 140–143]	
Image restoration	Uncond DM		Revising Diffusion Process		[138, 139, 144]	
	Uncond DM		Guidance		[146, 148, 153–156]	
	Uncond DM		Conditional Correction		[168, 169, 171]	
Image editing	Uncond DM / T2I DM [19, 20]		Inversion		[104–109, 111–115, 118–120]	
	T2I DM [19, 20]		Inversion, Conditional Correction		[170, 172, 173, 175, 216, 217]	
	T2I DM [19, 20]		Inversion, Attention Manipulation		[117, 121, 123–125, 129]	
	T2I DM [19, 20]		Inversion, Attention Manipulation, Guidance		[150, 162, 163, 166]	
Visual to image	T2I DM [19]		Guidance		[149, 159–161]	
Image composition	Uncond DM		Noise Blending		[134–136]	
Layout control	T2I DM [19, 23]		Attention Manipulation		[23, 102, 117, 125, 129, 172]	
	T2I DM [19, 20]		Attention Manipulation, Guidance		[67, 164, 165]	
General purpose	Unspecified		Noise Composition		[130]	
	Unspecified		Classifier-free Guidance		[131–133]	
	Unspecified		Universal Guidance Framework		[151, 152]	

for computational efficiency. In the training stage, the text-conditioned diffusion model $\epsilon_{\theta}(\mathbf{z}_t, t, \mathbf{c})$ is trained on this latent space to approximate the conditional distribution of the latent representations. In sampling process, the latent representation aligned with given text prompt is firstly generated by the conditional diffusion model on latent space, and then fed into the decoder to recover its corresponding high-quality image.

For conditional control, Stable Diffusion introduces a transformer text encoder to interpret the text prompt and convert into the text embedding. Subsequently text embedding is fused with the features in U-Net architecture of denoising network [19] via cross-attention mechanism. In practice, the encoder can be different domain-specific experts other than the text encoder. Thereby, Stable Diffusion can be employed

into various conditional synthesis tasks beyond text-to-image.

Following up the pioneer DM-based text-to-image framework GLIDE [21], Imagen [20] prefer to train the conditional denoising network on a low-resolution image space and subsequently upsample the synthesized low-resolution image. In order to effectively capture the complexity and compositionality of arbitrary text prompts, Imagen employs pre-trained large language models (e.g., BERT [219], GPT [220], T5 [221]) as powerful text-encoders. For condition injection, Imagen [20] concatenates the encoded text embedding to the key-value pairs of the self-attention layers in denoising network. In Imagen, the basic 64×64 text-to-image diffusion model is followed by two cascaded super-resolution diffusion models designed to enlarge the resolution of synthesized image from

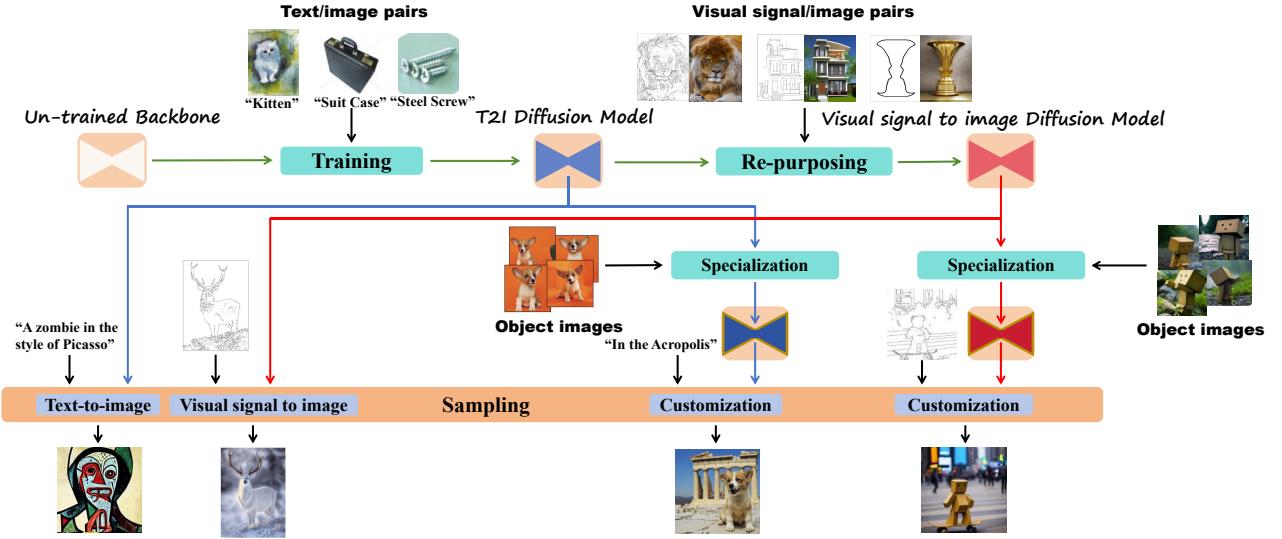


Fig. 3: An example of the workflow to build denoising network via training, re-purposing and specialization stages for target conditional synthesis tasks. In this framework, a text-to-image (T2I) denoising network is firstly obtained via supervised learning on text/image pairs in *training stage*. Subsequently, this T2I denoising network is fine-tuned on visual signal/image pairs for visual signal to image task in *re-purposing stage*. Next, both T2I and visual signal to image denoising networks can be further fine-tuned on given object image in *specialization stage* to perform customization on the user-specified personal object. Figures are cited from [19, 45, 47, 95].

64×64 to 1024×1024 .

2) *Conditional Models for Image Restoration*: DM-based conditional training is also widely employed to recover the high-quality clean image \mathbf{x} from a given degraded image \mathbf{c} [25–27, 30, 34]. These works primarily revolve around identifying the task-related features in degraded image as the conditional input for supervised training and recovering the clean image based on the model trained on these core features.

2.1) *Conditioning on degraded images*. The most straightforward modeling approach is directly conditioning the diffusion model on the given degraded image via channel-wise concatenation. Pioneer DM-based super-resolution method SR3 [25] concatenates the low-quality reference image with the latent variable \mathbf{x}_t in the channel space of U-Net architecture. This simple operation empowers the U-Net architecture to comprehensively capture information in low-resolution image. Concurrent SRdiff [28] shifts the generative space of SR3 to the residual space, and models the residuals between paired high and low resolution image to avoid regenerating the structures already existing in the low-resolution image. As a result, SRdiff performs on par with SR3 with significantly fewer computations. To adapt SR3 to real world restoration tasks, SR3+ [29] employs second-order degradation simulation to create real-world clean/degraded image pairs to enhance the training dataset. Based on SR3 [25], CDM [26] proposes to cascade super-resolution DMs to enlarge image resolution, and Palette [27] extends to more diverse image restoration tasks via supervised training on corresponding paired clean/degraded image datasets.

2.2) *Conditioning on pre-processed features*. However, simply concatenating the degraded image in the channel space places a burden on the denoising network to extract infor-

mation relevant to the restoration task from the unprocessed degraded image. To dedicate most modeling capacity on the task-related features, a branch of restoration works [30–34] prefer to firstly extract these features from the degraded image and subsequently conditioning the model on these task-related features.

State-of-the-art super-resolution framework Resdiff [30] employs a pre-trained CNN to generate a higher quality intermediate image for the initial degraded image, and conditions the denoising network on the intermediate image and its high-frequency details to synthesize the residual between the intermediate image and the clean image. For more complex restoration tasks including underwater image restoration [34] and low-light image enhancement [32, 33], in which the given degraded image is severely corrupted, a branch of works prefer to condition the model on frequency information extracted by discrete wavelet transformations. To restore real-world text images under severe degradation, DiffTSR [31] conducts parallel diffusion processes consist of an image diffusion model for image restoration and a text diffusion model for text recognition and employs a multi-modality module to interact the information of text and image diffusion process.

3) *Conditional Models for Other Synthesis Scenarios*: Although the mainstream DM-based frameworks for complicated conditional synthesis scenarios are established by re-purposing the text-to-image backbone, some works also prefer supervised training from scratch for different conditional synthesis tasks. Part of these works are early studies before the popularity of DM-based text-to-image models designed for tasks including image editing [35] and visual signal to image [36, 37]. Another part of these works are designed for novel or highly specialized conditional synthesis scenarios including medical

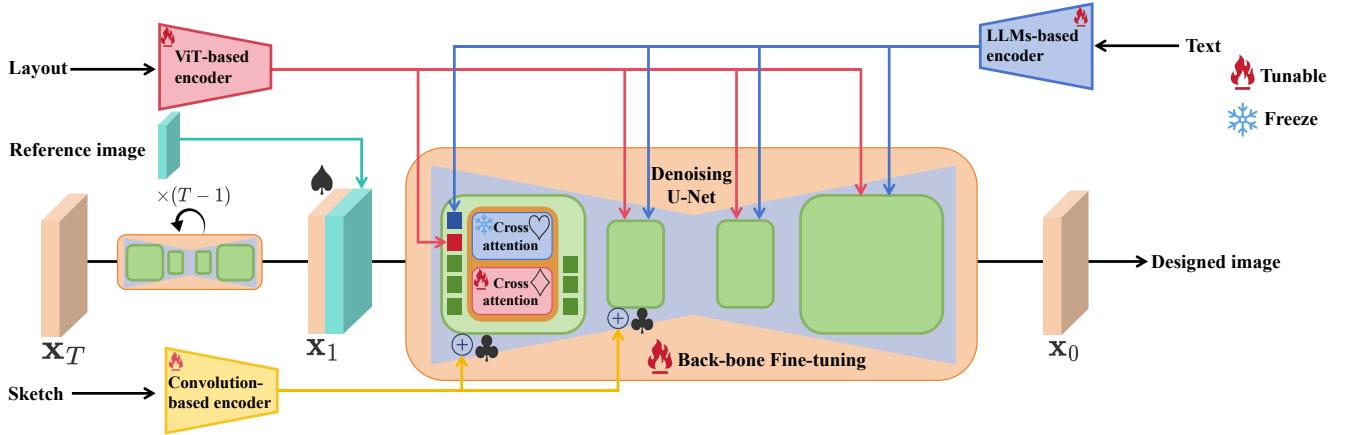


Fig. 4: An illustration of the re-purposed denoising network based on text-to-image backbone, where ♠, ♡, ♣, ♦ denotes condition injection via channel-wise concatenation, T2I attention layers, addition and developed attention modules respectively as described in Sec. III-B2.

image synthesis [38–41], graph-to-image [42] and satellite image synthesis [43], in which the conditional control signals are difficult to be aligned with the semantic space of the text-to-image backbone.

B. Condition Integration in the Re-purposing Stage

Currently, diffusion models (DMs) are employed in increasingly diverse and complex conditional synthesis scenarios [45, 65, 66, 70, 73, 74, 122]. Simply training denoising networks from scratch for each conditional synthesis scenario would place a heavy burden on computational resources. Fortunately, pre-trained text-to-image (T2I) DMs associate text embedding with its corresponding image, which serves as a semantic powerful backbone for a wide range of conditional synthesis tasks beyond the T2I. Studies design task-specific denoising network based on T2I backbone and performing fine-tuning on paired conditional inputs and image to re-purpose the base T2I denoising network to the target task. In practice, the re-purposed denoising network can be divided into three key modules: (a) *Conditional encoder*: The module to encode the task-specific conditional inputs into feature embedding, (b) *Condition injection*: The module to inject task-related feature embedding into T2I backbone, (c) *Backbone*: The T2I backbone that can stay frozen or be fine-tuned during the re-purposing stage. In the re-purposing stage, conditional fine-tuning can be performed in each of these components for condition integration as illustrated in Fig. 4. Subsequently, we will summarize the design choice for these modules among current works performing condition integration in the re-purposing stage.

1) *Re-purposed Conditional Encoders*: In a T2I model, the text embedding is extracted from the given text prompt through a text encoder and subsequently injected into the U-Net architecture through cross-attention layers. To re-purpose the T2I backbone to tasks beyond text-to-image, various task-specific conditional encoders are designed to extract the features from conditional control signals other than text.

1.1) *Convolutional layer-based encoder for visual signals*. For visual signals, conditional encoders are mainly designed base on convolutional downsample blocks to extract multi-scale structure features.

Pioneer work T2I-Adapter [44] employs a four-layer convolutional network as a lightweight adapter to encode the visual signal into a set of multi-scale features. ControlNet [45] provides a more powerful architecture as the encoder for visual signals, which clones the deep encoding layers from the U-Net architecture in Stable Diffusion [19]. This ControlNet encoder inherits a wealth of prior knowledge in the Stable Diffusion backbone and serves as a deep, robust, and strong architecture for diverse visual signals. Currently, ControlNet delivers state-of-the-art results in diverse visual signal to image tasks and becomes a the widely-employed conditional encoder various more complicated conditional synthesis scenarios including explicit lighting control [49], image composition [50], image editing [51, 87] and virtual try-on [213, 214].

1.2) *ViT-based encoder for images*. In practice, Vision Transformer (ViT)-based encoders are widely employed to extract features from conditional inputs in form of image. Generally, visual signals can also be viewed in form of image, the pioneer work PITI [46] designs a ViT-based encoder to map the visual signal into its corresponding text embedding for the T2I backbone. ImageBrush [52] also employs a ViT-based encoder to extract the visual editing instruction described by paired images before/after editing. Prompt-free Diffusion [53] employs a more powerful context encoder based on SWIM-L [222] to convert image into visual embedding. For customization, a branch of works [47, 54–61] maps the given personal object into features on the textual space via different ViT-based image encoders based on the framework of CLIP [220], SWIN [222], BLIP [223] or ViT-based ArcFace encoder [224].

1.3) *LLMs-based encoder for image editing*. In order to enhance the semantic information in the given text prompt, a branch of works prefer to design more powerful Large Language Models (LLMs)-base encoders for text-based image

editing, [48, 63, 64] leverages a trainable Multimodal Large Language Models (MLLMs) [225] module as the encoder for the given source image and the editing instruction. Ranni [62] uses LLMs to convert description or editing prompts into a semantic panel, which serves as an intermediate representation that contains rich structure and semantic information.

2) *Condition Injection*: In order to more effectively incorporate information from the conditional input into the denoising network during the re-purposing stage across various conditional synthesis scenarios, studies have developed different task-specific condition injection approaches to handle different types of conditional control signals. Here, we categorize these methods into the following four categories.

2.1) *Condition injection via concatenation ♠*. For conditional inputs in form of image, a direct condition injection approach is following the concatenation strategy proposed by SR3 [25], which concatenates the image form conditional inputs to the latent variable \mathbf{x}_t in the channel space of the U-Net architecture. In practice, this conditioning strategy is usually performed with backbone fine-tuning to handle conditional synthesis tasks that involve complex conditional inputs composed of multimodal components, including instruction-based editing [73, 78, 79] and image composition [74, 82, 84].

2.2) *Condition injection via T2I attention layers ♥*. In the T2I backbone, the cross-attention layers serve as the conditioning module to inject text embedding into the U-Net architecture. Currently, a branch of works also employ the cross-attention layers in T2I backbone to inject the features extracted from task-specific conditional encoders [46, 47, 52–54, 56, 57, 61, 213].

2.3) *Condition injection via addition ♣*. Because of the alignment between the architecture of conditional encoder and the U-Net encoder in T2I backbone, for convolutional layer-based encoders [44, 45], the extracted features are injected via directly adding these features to the corresponding intermediates layers of U-Net architecture in T2I backbone.

2.4) *Condition injection via developed attention modules ♦*. To achieve more fine-grained control over the synthesized image, some works design developed task-specific attention modules for condition injection in target conditional synthesis scenarios [65–68, 70].

A branch of works prefer to incorporate extra attention module into the T2I backbone to inject the task-specific conditional control signals [65, 66, 68–70]. IP-adapter [65] employs additional image cross-attention layers to inject the image embedding into the T2I backbone. For customization, ELITE [68] leverages two parallel cross-attention layers to inject extracted global and local information of given personal object separately. In T2I backbone, attention layers control the structure and layout information of synthesized image. To exert accurate object-level layout control, a branch of works prefer to add a trainable attention-module between self-attention and cross-attention layers [55, 60, 66, 69, 70]. GLIGEN [66] adds a gated self-attention layer to U-Net architecture to inject provided layout information. This conditioning strategy is further employed in customization works [55, 60] to integrate patch features extracted from personal object images. To perform more detailed layout control, InteractDiffusion [69]

designs an attention-based Human-Object Interaction module to inject the interactions between objects. InstanceDiffusion [70] projects different forms of object-level control signals including single points, scribbles, bounding boxes or intricate instance segmentation masks into the feature space through a UniFusion block, and inject these features with a Instance-Masked Attention module.

Another line of works modify the cross-attention mechanism in T2I backbone to achieve more precise control [59, 67, 71, 72]. Different from IP-adapter [65], DEADiff [71] concatenates the key and value features from image and text embedding respectively and perform a single fused cross-attention mechanism to achieve multimodal conditional control. In practice, performing fused attention mechanism to inject multimodal control signals along with text embedding is also employed in instruct-based editing [207] and pose-guided person image synthesis [59]. To perform local control based on multiple regional prompts, Mix-and-show [72] proposes an attention localization strategy in the re-purposing stage, which substitutes the attention map in specified regions with the attention map generated based on the regional prompts.

3) *Backbone Fine-tuning*: Currently, most of the re-purposing works confine the fine-tuning only on conditional encoders and condition injection modules to ease the computational burden. However, for conditional inputs containing multimodal components or intricate semantics, performing fine-tuning while freezing the parameters in T2I backbone often fails to fully learn intrinsic connections between the conditional input and target image. In this case, fine-tuning the T2I backbone together with conditional encoders and condition injection modules is a more preferable choice. Base on the fine-tuning strategy, we categorize these works into two types: (a) Fully supervised fine-tuning on annotated datasets, and (b) Self-supervised fine-tuning on bare image datasets.

3.1) *Fully supervised fine-tuning on annotated datasets*. In practice, we can re-purpose the T2I backbone on the annotated dataset of paired conditional input and image in accordance with the target task via fully supervised fine-tuning. For some synthesis tasks involving complex conditional inputs, a major difficulty lies in collecting sufficient training data to fine-tune the model [73, 74]. For instruct-based editing task which refers to using instruction instead of text description to guide the editing process, Instructpix2pix [73] provides an effective approach for automatically synthesizing training datasets. Firstly, InstructPix2Pix employs a fine-tuned GPT-3 [226] to synthesize editing triplets composed of input captions, edit instructions and output captions. Subsequently, Instructpix2pix leverages Prompt-to-Prompt [123] to synthesize paired images corresponding to the input captions and output captions, which serves as the paired images before/after editing. This contribution leads to a line of works on DM-based instruction editing. A branch of follow-up works attempt to enhance the model in some specific tasks by augmenting the training dataset for target scenario including object removal and inpainting [75], global editing [207], dialog-based editing [76] and continuous editing [77]. InstructDiffusion [78] and Emu-edit [79] fine-tune the T2I backbone on larger and more comprehensive synthesized datasets for a wide range of

vision tasks including image editing, segmentation, keypoint estimation, detection, and low-level vision. To achieve more accurate editing, [48, 63, 64] fine-tune the T2I backbone with more powerful MLLMs-based conditional encoders to enhance the editing prompts. Based on reinforcement learning, HIVE [80] fine-tunes the instruct-based editing model with a reward model reflecting the human feedback for editing performance.

3.2) Self-supervised fine-tuning on bare image dataset. In non-general conditional synthesis scenarios involving image composition or mask-based editing, the form of conditional inputs may be complicated. For example, a classic image composition task aims to fuse a foreground reference image into the background main image within the mask region. In these tasks, collecting annotated training data pairs is almost impossible. A feasible approach is to create paired data for the target scenario through cropping on a bare image dataset, and fine-tune the T2I backbone in a self-supervised manner. For image composition task, PbE [74] randomly crops the foreground objects from the source image as the reference image and corresponding mask, while the remained background as the background main image. Subsequently, PbE [74] fine-tunes the T2I backbone with paired cropped reference image and main image. In practice, such strategy is widely employed in conditional synthesis scenarios involve inpainting [81, 82] and image composition [50, 83–86]. To generate reasonable masks for text-based inpainting, Imagen Editor [81] employs an off-the-shelf object detector to generate mask on the image in captioned image datasets, which covers a region relevant to the text caption of image. SmartBrush [82] randomly augments the cropped training masks to create accurate instance masks, which facilitates the T2I backbone to follow the shape of the input mask at testing-time.

For image composition, the greatest challenge faced by the self-supervised fine-tuning strategy is how to avoid the trivial copy-and-paste solution caused by the training data cropped from a single image [74, 83, 87]. Currently, image composition works resort compress the information in the conditional inputs into an information bottleneck. This, in turn, forces the T2I backbone to interpret the intrinsic connections between the conditional input and the desired image, thereby effectively avoiding the copy-and-paste solution. PbE [74] and Dream-inpainter [83] select part of the image tokens for condition injection to create information bottleneck. ObjectStitch [84] employs a two-stage fine-tuning strategy to decouple the fine-tuning stages of the conditional encoder and the T2I backbone. [50, 86, 87] prefer to remove or mask out the information such as colors, textures or background in source image to prevent identical mapping.

C. Condition Integration in the Specialization Stage

Although theoretically we can incorporate any form of conditional inputs \mathbf{c} into the denoising network $\epsilon_\theta(\mathbf{x}_t, t, \mathbf{c})$ during the training and re-purposing stages, for complicated conditional synthesis scenarios, incorporating such control signals into the conditional space of denoising network faces challenges in collecting annotated training dataset and modeling the complicated correlation between conditional inputs

and desire results. This limits the model capability to deal with zero-shot or few-shot conditional inputs.

A straightforward idea to remedy these issues is to align the given conditional inputs with the conditional space of a general T2I backbone through a specialization stage. As shown in Fig. 5, the specialization for given specific conditional inputs is typically achieved by (a) *conditional projection*, which projects the given conditional inputs onto the conditional space of the T2I backbone via embedding optimization [88, 89], or Vision-Language Pre-training (VLP) frameworks [223, 227], (b) *testing-time model fine-tuning*, which fine-tunes the denoising network to insert the conditional inputs into the prior of the T2I backbone. In practice, works perform condition integration in specialization stage are mainly targeted to image editing and customization tasks to achieve desired edits on user-specified visual subjects including source images(image editing) and personal objects(customization) while preserving the characteristics and details in these visual subjects [88, 89, 95].

1) Conditional Projection: To perform editing or customization tasks, a widely employed approach is projecting the given visual subject into its corresponding text representation on the conditional space of text-to-image model.

1.1) Conditional embedding optimization. In order to find a proper text embedding for given visual subject, a branch of works directly search for the optimal embedding for the user-specified conditional inputs by optimizing the following objective function:

$$\mathbf{v}^* = \arg \min_{\mathbf{v}} \mathbb{E}_{\mathbf{x}=\mathbf{c}_I, \epsilon, t} \left[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t, \mathbf{v})\|_2^2 \right], \quad (7)$$

where \mathbf{v}^* denotes the optimized text embedding for the user-specified visual subject \mathbf{c}_I , and ϵ_θ denotes the T2I backbone. The embedding \mathbf{v}^* serves as a pseudo-word S^* for the visual subject and can be further composed into various natural language prompts to create different editing renditions for given visual subject [88, 89].

For image editing, Imagic [88] optimizes the embedding \mathbf{v}^* for the source image. Subsequently, Imagic performs interpolation between optimized source embedding \mathbf{v}^* and target embedding \mathbf{v}_{tgt} to obtain $\bar{\mathbf{v}} = \eta \cdot \mathbf{v}_{tgt} + (1 - \eta) \cdot \mathbf{v}^*$, which serves as the conditional input for denoising network. Diffusion Disentanglement [90] optimizes the time-specified combination weights $\lambda_{1:T}$ of the source and target text embedding along the sampling process instead of interpolation to retrieve time-adaptable embedding for editing. To reduce the computational cost of the optimization process, [93, 166] first employ image encoder to generate a coarse embedding of the given visual subject, and subsequently fine-tuning the coarse embedding via optimization.

Pioneer customization work Textual inversion [89] perform optimization to discover the text embedding \mathbf{v}^* for personal object described by a few reference images (typically 3 to 5). This optimized embedding \mathbf{v}^* serves as the pseudo-pronoun S^* for the personal object in further conditional sampling process. To provide human-readable text description instead of text embedding for the given personal object, PH2P [91] employs quasi-newton L-BFGS [228] to directly optimize

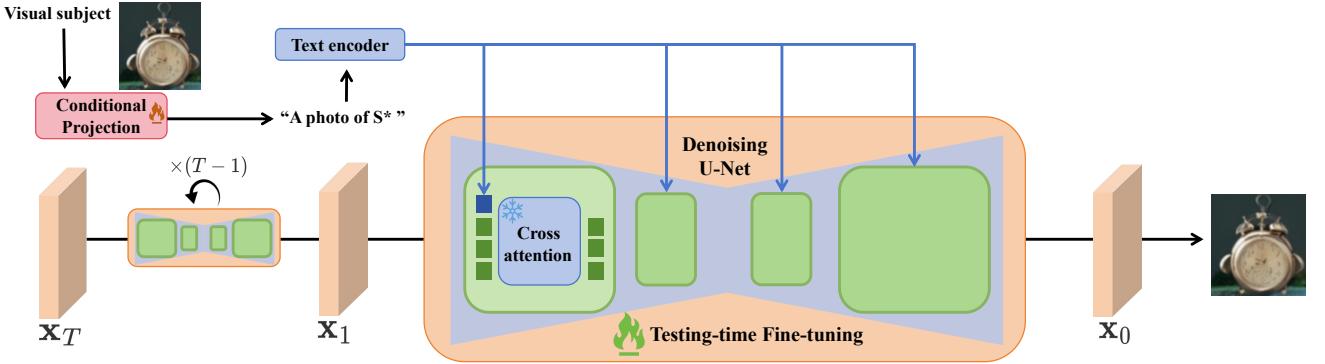


Fig. 5: The specialization stage to align a given personal object (the clock) with a pesudo-word S^* in the conditional space of a text-to-image backbone. The clock image is cited from Textual Inversion [89].

discrete tokens from a existing pre-specified vocabulary for the given image.

1.2) Employing VLP models. However, performing time-consuming optimization process for each new visual subject hinders the deployment of these methods in application scenarios. Therefore, a branch of works prefer to employ Vision-Language Pre-training (VLP) models to directly generate the embedding for given visual subjects [47, 93].

BLIP [227] is a strong VLP framework to synthesize captions for given images, which is widely employed in image editing tasks to generate an initial text prompt to describe the uncaptioned source image [47, 93, 94, 150]. BLIP can also be used to enhance user-provided prompts for eliminates editing failure caused by missing contexts in the coarse input prompts [229]. Besides, PRedItOR [92] prefers to leverage DALL-E2 [22] to fuse the source image with the target prompt by performing SDEdit [104] process on the CLIP embedding space.

2) Testing-time Model Fine-Tuning: In editing and customization tasks, simply employing the denoising network modeled in scenario-orient training and re-purposing stages always fails to retain the characteristics and details in the user-specified visual subject, due to the lack of prior knowledge [99]. To customize the T2I backbone for the user-specified conditional input, approaches in this category resort to perform testing-time fine-tuning on the T2I backbone to insert the given visual subjects into the denoising network [95, 99].

To better preserve the outlook of source image in editing tasks, a branch of works [88, 93, 96, 98] represented by Imagic [88] fine-tune the T2I backbone to bind the source image with its corresponding text description c_{src} in the conditional space. In order to simultaneously editing the foreground and background in the source image, LayerDiffusion employ Segment Anything Model (SAM) [230] to create masks for foreground objects. Subsequently, LayerDiffusion [97] fine-tunes the T2I backbone with a designed loss composed of the diffusion loss in both foreground and background region to editing the foreground object and background independently. SINE [98] introduces a patch-based fine-tuning strategy which incorporates the positional embedding into conditional T2I space to synthesize arbitrary-resolution edited image.

For the customization task, DreamBooth [95] fine-tunes the T2I backbone to entangle a fixed unique identifier with the semantic meaning of the personal object. To alleviate the computational burden in the testing-time fine-tuning, followed up works [56, 72, 99–103] prefer to only fine-tune a specific part of model parameters. CustomDiffusion [99] fine-tunes only the cross-attention layers. E4T [56] optimizes low-rank adaptations (LoRA) [231] of weight residuals in cross- and self-attention layers to further reduce computational cost. Cones [101] fine-tunes the attention layer concept neurons highly-related to the given visual subject. Cones2 [102] and Mix-and-show [72] resort to fine-tune the text encoder in T2I backbone. SVDiff [103] fine-tunes the singular values of the decomposed convolution kernels.

IV. CONDITION INTEGRATION IN THE SAMPLING PROCESS

In DM-based image synthesis frameworks, the sampling process iteratively reserve noisy latent variable into desired image with the prediction of the denoising network. As mentioned in Sec. III, integrating the conditional control signals into the denoising network always requires time-consuming training, fine-tuning or optimization. To ease the burden for conditioning the denoising network, numerous works perform condition integration in the sampling process to ensure the consistency between synthesized image and given conditional input without computational intensive supervised-training or fine-tuning [105, 123, 130, 139, 145, 169].

Based on how the conditional control signals are incorporated into the sampling process, we divide mainstream insampling conditioning mechanisms into six categories: (a) *inversion*, (b) *attention manipulation*, (c) *noise blending*, (d) *revising diffusion process*, (e) *guidance* and (f) *conditional correction*. We illustrate these conditioning mechanisms with an exemplary image editing process in Fig. 6. In this section, we will introduce the core idea of these conditioning mechanisms and summarize the corresponding representative works.

A. Inversion

In diffusion model (DM)-based image synthesis, the starting latent variable controls the spatial structure and semantics of

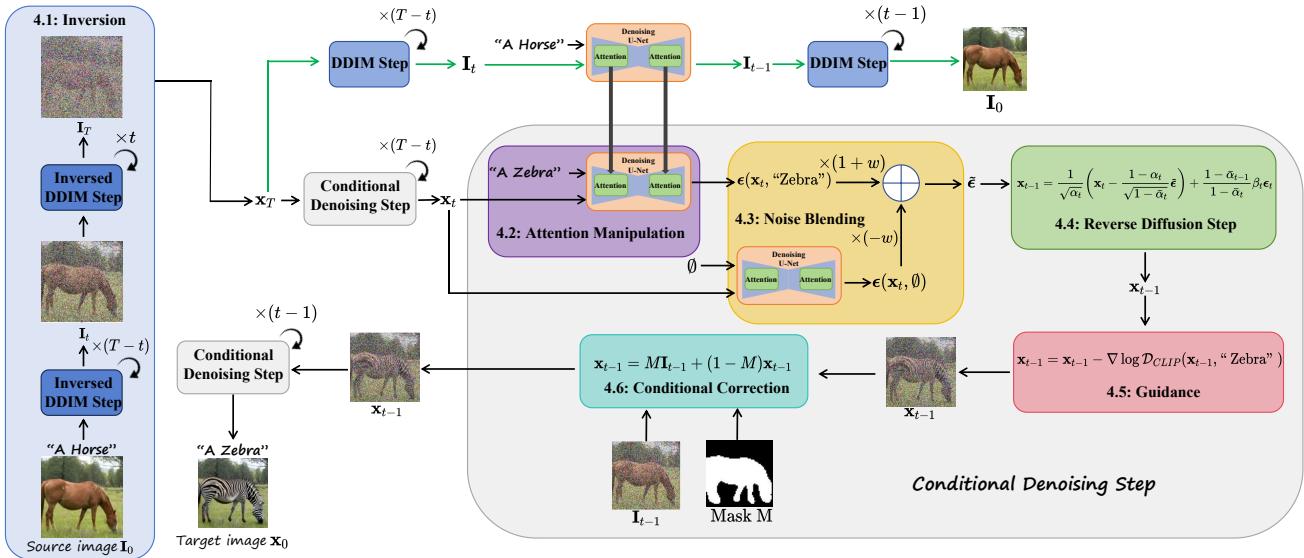


Fig. 6: An example of the conditional sampling process for image editing, in which we incorporate all six mainstream in-sampling conditioning mechanisms for sampling process to provide a comprehensive overview of the content in this section. The sample images are from Diffedit [170].

synthesized result. Inversion process provides an effective way to encode the given source image back into its corresponding starting latent variable and effectively preserve the image structure and semantics for further editing. In this section, we firstly summarize the inversion approaches in Sec. IV-A1. Next, we will discuss the applications of inversion in various conditional synthesis scenarios in Sec. IV-A2.

1) Inversion Approaches: Mainstream inversion approaches perform inversion based on the forward diffusion process, deterministic sampling process, and stochastic sampling process. We denote these three basic inversion pathways as *noise-adding inversion*, *deterministic inversion*, and *stochastic inversion*, respectively. Due to accumulated errors in the discrete diffusion process, the naive inversion process often fails to preserve details in the source image, especially with classifier-free guidance. Therefore, numerous works propose enhancements to these basic inversion approaches to ensure perfect reconstruction of the source image.

1.1) Noise-adding inversion. Noise-Adding Inversion performs a standard forward diffusion process to inverse the source image to a certain noise step T' , i.e., $q(\mathbf{x}_{T'} | \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{T'}; \sqrt{\bar{\alpha}_{T'}} \mathbf{x}_0, (1 - \bar{\alpha}_{T'}) I)$, where the latent variable $\mathbf{x}_{T'}$ is a mixture of source image and Gaussian noise.

1.2) Deterministic inversion. However, noise-adding inversion may smooth out details in the source image. To more precisely preserve image features, deterministic inversion is proposed to encode the source image \mathbf{x}_0 into its corresponding latent variable \mathbf{x}_T with the discretization of diffusion ODEs such as DDIM [187]. Theoretically, with a sufficiently large diffusion step T , DDIM inversion can guarantee perfect reconstruction, which ensures the latent variable \mathbf{x}_T obtained from DDIM inversion to be a meaningful diffusion starting point encapsulating all features pertaining to the source image \mathbf{x}_0 .

1.3) Stochastic inversion. However, DDIM inversion performs accurate inversion only when the diffusion time steps

is sufficiently large, which always leads to unsatisfied results especially under classifier-free guidance. Therefore, a branch of works prefer to inverse the stochastic sampling process in Eq. 2. Different from the deterministic sampling process, which is determined by the starting point latent variable \mathbf{x}_T , the stochastic sampling process involves the noise vector ϵ_t added in each reverse transition kernel. Therefore, we have to memorize each noise vector ϵ_t along the inversion process to ensure the reconstruction property.

1.4) Enhanced inversion approaches. In conditional synthesis, the classifier-free guidance significantly magnified the accumulated error in inversion process, which leads to poor reconstruction and edit performance. Therefore, a series of inversion methods are developed to ensure the inversion performance under classifier-free guidance.

For deterministic inversion, some approaches prefer to *fine-tune* relevant parameters in the classifier-free guided sampling process to reduce the reconstruction error, including optimizing the null-text embedding [106], text embedding for the source image [109], key and value matrix in the self-attention layers [110], and the prompt embedding for cross-attention layers [111]. To get rid of the computational burden for fine-tuning, a branch of works has developed *tuning-free* approaches for perfect reconstruction [112–114, 232]. EDICT [112] achieves precise DDIM inversion by utilizing an equivalent reversible process consisting of two coupled noise vectors. Negative-prompt Inversion [113] demonstrates the prompt of the source image can serve as a training-free substitute for null-text embedding. Proxedit [232] further enhances the reconstruction performance of Negative-prompt Inversion [113] by incorporating a regularization term in classifier-free guidance to prevent over-amplifying the editing direction in sampling process. Fixed-point Inversion [115] and AIDI [116] perform fixed-point iterations in each step of DDIM inversion to reduce the accumulation errors due to the

discrete DDIM process. Besides, Fixed-point Inversion [115] provides a brief cycle of fixed-point iterations for the VAE-encoded latent representation of source image to eliminate the misfit between latent representation and given text prompt in latent diffusion model. TF-ICON [117] and LEDITS++ [118] perform inversion based on high-order diffusion differential equation solvers [233, 234] which significantly accelerates the inversion process and improve the accuracy of inversion.

For stochastic inversion, theoretically, any sampling sequence starts with source image can be employed as the iterative latent variables in stochastic inversion process. However, arbitrary sampling sequence will deviate from the prior marginal distribution of latent variables and harm the editing ability in reconstruction process. To construct a reasonable sampling sequence, pioneer work Cyclediffusion [107] firstly samples a $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$ and subsequently denoise it based on the source image \mathbf{x}_0 to recover the sampling sequence. DDPM inversion [108] constructs an editing-friendly sequence by sampling each intermediate latent variable \mathbf{x}_t independently based on the source image \mathbf{x}_0 and reconstructs the source image up to noise precision to avoid error accumulation. SDE-Drag [119] provides a theoretical fundamental to explain the superiority in editing performance of stochastic inversion comparing to deterministic inversion. It demonstrates that the KL-Divergence between the distribution of edited image and prior data distribution decrease in stochastic inversion while remaining in widely used deterministic inversion.

2) *Applications of Inversion in Conditional Synthesis:* Inversion process converts the provided source image into its corresponding latent variable. In practice, this latent variable can serve as the starting point for sampling process to perform basic image-to-image translation, text-based image editing or be further manipulated for more complicated tasks.

Image-to-image translation target to translate the content in a given source image into the desired appearance, which serves as the foundation for image editing. Pioneer work SDEdit [104] translates a given out-of-domain source image into its counterpart in target domain by denoising the noise-adding inverted source image with the denoising network trained on target domain. This process preserves the content in source image while endowing it with appearance in the target domain.

Based on deterministic inversion, DDIB [105] introduces a highly flexible technique for image-to-image translation between two manifolds α and β via a simple process $\mathbf{x}^* = \mathcal{D}_\beta(\mathcal{E}_\alpha(\mathbf{x}))$, where \mathbf{x} and \mathbf{x}^* denote the source and target image on manifold α and β respectively, \mathcal{E}_α and \mathcal{D}_β denote the deterministic inversion and sampling process performed with the diffusion models for manifold α and β . DDIB process can be performed with two independently trained diffusion models or a diffusion model conditioned on different control signals.

In practice, text-based editing task, which targets to edit the source image \mathbf{c}_I described by \mathbf{c}_{src} to align with target text prompt \mathbf{c}_{tgt} , can be achieved by performing the DDIB image-to-image translation process as $\mathbf{x}^* = \mathcal{D}_{\mathbf{c}_{tgt}}(\mathcal{E}_{\mathbf{c}_{src}}(\mathbf{c}_I))$, where $\mathbf{c}_I, \mathbf{x}^*$ are paired source and edited image, and $\mathcal{D}_{\mathbf{c}_{tgt}}$ and $\mathcal{E}_{\mathbf{c}_{src}}$ denotes the sampling process conditioned on target prompts and the inversion process conditioned on source prompts.

However, this editing process can only roughly ensure the consistency in semantics and overall structure while always failing to precisely preserve the intricate details in source image. In order to more accurately recover the details in source image in editing process, inversion is always performed with other conditioning mechanisms in the editing process. Performing conditional correction with mask is a preferable choice to preserve the region not requiring editing [97, 127, 170, 172–175]. Another choice is performing attention manipulation during the editing process to incorporate the outlook of source image, as discussed in Sec. IV-B [123, 124]. Besides, a branch of works employ model fine-tuning in specialization stage or conditional projection described in Sec. III-C to inject the detailed outlook of source image into the T2I backbone [88, 93].

Besides, based on the task-specific conditional encoders to convert multi-model conditional inputs into text embedding, this =inversion-based editing process can also be employed in conditional synthesis tasks beyond text-based editing. For example, InST [120] denoises the noisy reference image obtained by noise-adding inversion with the denoising network conditioned on the embedding vectors extracted from the style image to achieve style transfer editing.

For more complicated conditional synthesis scenarios, the latent variable obtained from inversion can be manipulated to incorporate additional information beyond the source image. For image composition, a branch of works prefer to fuse the latent variable obtained from inversion process for different source images [117, 121]. Style Injection in Diffusion [121] fuses the latent variable of both style and content image obtained by DDIM inversion to perform style transfer. TF-ICON [117] composes the inverted main and reference images for image compositing. In drag-based editing, we can adjust the corresponding area in the latent variable based on the provided drag instructions. Dragdiffusion [122] optimizes the latent variable with designed motion supervision loss for drag-style manipulation. The stochastic inversion-based work, SDE-Drag [119], manipulates the latent variable through a copy-and-paste strategy instead of performing optimization in the latent space.

B. Attention Manipulation

After determining the starting point for the sampling process via sampling from Gaussian distribution or inversion methods, the sampling process is performed by iterative denoising steps. As pointed out in E4T [56], the attention layers in the denoising network have the greatest influence on the predicted noise in each denoising step and thereby control the structure and layout of synthesized image. Therefore, a branch of works resort to design task-specific manipulation to the attention layers in denoising network to achieve more accurate control over the spatial layout and geometry [117, 123, 124, 172]. Different from the works [65, 66] performing fine-tuning on modified attention module in re-purposing stage, approaches in this category manipulates the attention layers via tuning-free replacement or localization during sampling process.

1) Replacement Manipulation: Pioneer attention manipulation works are designed to preserve the structure of source image during the inversion-based image editing process. Prompt-to-Prompt [123] performs parallel sampling processes for the inverted source image separately conditioned on source and target prompts. During the parallel sampling process, Prompt-to-Prompt replaces the cross-attention maps in editing branch with its counterpart in reconstruction branch in order to preserve the structure of source image during the editing sampling process. This replacement strategy is further employed in followed up works for face aging editing [126] and customization-based editing [100]. P2Plus [111] further replaces the editing branch self-attention map in the unconditional noise predictor network with its counterpart in reconstruction branch to obtain more accurate editing capabilities with classifier-free guidance. In order to prevent undesired changes caused by cross-attention leakage, DPL [127] optimizes the word embedding corresponding to the noun words in source prompt to produce more suitable cross-attention maps for attention replacement.

PnP [124] points out that more detailed spatial features are restored in self-attention layers comparing to cross-attention maps. Therefore, a branch of editing works [124, 125, 128] prefer to replace query and key feature in self-attention layer to achieve better structure preservation. This replacement strategy is followed by works for drag-based editing [67, 122] and style transfer [121] to ensure the consistency between synthesized result and provided source image.

2) Attention Localization: To achieve more precise layout control for the synthesized image, a branch of works manipulate the attention layers with masks or segmentation indicating the locations of objects [23, 117, 172].

Some of these works propose localized self-attention mechanisms to address different regions separately and locate the contents into desired regions. Masactr [125] and Object-Shape Variation [172] firstly extract the regions with attention value above a threshold in the cross-attention maps for object text tokens as foreground masks. Subsequently, Masactr performs self-attention for foreground and background separately to prevent confusion between the foreground objects and the background. Object-Shape Variation [172] restrict the region for attention replacement on the background not requiring editing instead of injecting the full self-attention maps in every denoising step. For image composition, TF-ICON [117] fuses the attention features extracted from the reconstruction process for the reconstruction branches of both main and reference images via cross-attention mechanism to create a composite self-attention map seamlessly blending the two images.

Another line of works incorporate an increment into the cross-attention map to adjust the attention values in the region for designated objects and thereby achieve layout control for synthesized image. Pioneer text-to-image work Ediff-i [23] successfully guides the object described by the nouns in the text prompt to the specified area by enhancing the attention values in the corresponding region. Similarly, Cones2 [102] increases the attention values in the region corresponding to desired objects while reducing the attention values in irrelevant regions to perform layout control. For image editing, FoI [129]

amplifies the attention value in the region of foreground object to be edited to achieve more precisely control for the objects in accordance with editing instructions.

C. Noise Blending

Noise blending process fuses noises predicted by different (conditional) DMs to perform single sampling process controlled by multiple conditional signals.

1) Noise Composition: In conditional synthesis scenarios aiming at synthesizing images conditioned on multiple control signals, directly training a denoising network to take all conditional inputs always leads to an unsustainable training cost. A widely employed approach to tackle these tasks is predicting the noise ϵ_i for each conditional component c_i separately and subsequently composing these noise to acquire a novel proxy noise $\tilde{\epsilon}$ controlled by all the conditional control signals without supervised-learning. Composable Diffusion Models [130] present a noise composition approach based on Bayes' formula as follows to perform multi-conditional synthesis:

$$\tilde{\epsilon} = \epsilon_\theta(\mathbf{x}_t, t) + \sum_{i=1}^n w_i (\epsilon_\theta(\mathbf{x}_t, t, c_i) - \epsilon_\theta(\mathbf{x}_t, t)), \quad (8)$$

where the unconditional denoising network $\epsilon_\theta(\mathbf{x}_t, t)$ can be trained along with the conditional model by substituting the conditional parameter with emptyset \emptyset .

The noise composition can be performed based on masks or layouts to locate the objects in provided conditional inputs into desired regions. To perform image editing on multiple instructions, LEDITS++ [118] calculates the mask for the region related to each instruction with the grounding information in cross-attention layers and noise estimations. Subsequently, LEDITS++ [118] performs noise composition based on the formula of Eq.8 while restricting effect of the conditional term $\epsilon_\theta(\mathbf{x}_t, t, c_i) - \epsilon_\theta(\mathbf{x}_t, t)$ of each editing instruction c_i in its corresponding mask region. In order to fuse the generated results of two diffusion models, MagicFusion [134] firstly generates mask by contrasting the saliency map of the two diffusion models to differentiate the region controlled by each model. Subsequently, MagicFusion [134] settles the noise into the region controlled by its corresponding diffusion model. Similarly, NoiseCollage [135] independently estimates the noises for each individual object and then merges them with a crop-and-merge operation based on the provided layouts. In order to perform more seamless noise composition, Multi-diffusion [136] blends the noise by solving an optimization objective with closed-form optimal solution, which ensure the consistency of composed noise map $\tilde{\epsilon}$.

2) Classifier-Free Guidance: As described in Sec.IV-E, in traditional guidance, adjusting the guidance strength scaling factor w allows us to effectively balance the quality and diversity of synthesized samples. However, estimating the likelihood term $p_t(c | \mathbf{x}_t)$ in traditional guidance is challenging.

Classifier-free guidance [131] provides a new pathway to achieve balance the quality and diversity of synthesized samples without likelihood estimation, which can be achieved by performing extrapolation blending between the conditional noise prediction and the unconditional noise prediction as:

$\tilde{\epsilon}_\theta(\mathbf{x}_t, \mathbf{c}) = (1 + w)\epsilon_\theta(\mathbf{x}_t, \mathbf{c}) - w\epsilon_\theta(\mathbf{x}_t)$. In this formula, the parameter w controls the strength of guidance and the trade-off between sample quality and diversity. In practice, setting the scaling factor w to a value greater than zero can significantly enhance the sample quality and the consistency to the conditional control signal \mathbf{c} . In order to alleviate the negative impact of classifier-free guidance on sample diversity, followed works [132, 133] propose dynamic classifier-free guidance, in which the guidance scaling factor w is reduced during the denoising process with high noise levels.

Moreover, some works also propose variations of classifier-free guidance for different conditional synthesis scenarios. Instructpix2pix [73] and Pair diffusion [51] develop the classifier-free guidance to adjust the conditioning strength for each component in multiple conditional inputs by decomposing the multi-conditional score function. For customization tasks, SINE [98] interpolates the noise prediction on specialized and pre-trained model to obtain conditional noise prediction in classifier-free guidance, which alleviates the overfitting in the specialized model. Null-text Guidance perturbs the classifier-free guidance by altering the noise-level in unconditional prediction to smooth out some realistic details and create cartoon-style images. For inversion-based editing, AIDI [116] proposes a blended classifier-free guidance based on the positive/negative masks indicating the area to be edited or preserved, which enables larger guidance scales and ensures more accurate editing results.

D. Revising Diffusion Process

Most of in-sampling conditioning mechanisms such as Guidance, Conditional Correction and Attention Manipulation performs modification on the standard formulation of the denoising step, which leads to deviations from the predetermined sampling trajectory and results in artifacts in synthesized images. Therefore, a branch of works prefer to incorporate the conditional control signals into the denoising step via revising the formulation of standard diffusion process to adapt the conditional synthesis task [137, 139, 141, 144]. Thereby, the conditional control signals can be incorporated into the corresponding reverse diffusion step of the revised diffusion process without deviations from the diffusion formulation.

Based on the revision on diffusion process, these works can be divided into two categories: (a) *mean-reverting SDEs*, which revise the diffusion process to preserve the information in conditional inputs in image restoration, (b) *decomposition-based noise redefinition*, which incorporate a sequence of additive noises in the sampling process on spectral space to revise the noise-level mismatch in noisy linear problem.

1) *Mean-Reverting SDEs*: In numerous restoration tasks, most structure and semantic features of the target image is provided by the degraded image \mathbf{c} . To avoid consuming part of the model capability on regenerating these features from pure Gaussian noise, some studies design novel diffusion process in which the diffused output \mathbf{x}_T approximates a noisy version of degraded image \mathbf{c} instead of pure Gaussian noise [137, 140–143]. IR-SDE [137] construct a set of mean-reverting SDEs identified by degraded image \mathbf{c} , which models the diffusion

process from clean image \mathbf{x} to a Gaussian distribution averaged on degraded image. Subsequently, IR-SDE trains a conditional denoising network to predict the score function in the reversed mean-reverting SDEs to recover the clean image from the noisy degraded image. Similarly, ResShift [141] and DriftRec [140] construct an iterative degradation process from a high-resolution image to its corresponding low-resolution image as diffusion process and train a conditional denoising network to reverse the degradation process for super-resolution. SinSR [142] distills the sampling process of ResShift [141], thereby achieving one-step DM-based super-resolution. InDI [143] constructs a continuous forward degradation process derived from interpolation: $\mathbf{x}_t = (1-t)\mathbf{x} + t\mathbf{c}$ and trains a denoising network on paired clean/degraded image to predict clean image \mathbf{x}_0 from latent variable \mathbf{x}_t . Subsequently, image restoration can be performed by reversing the interpolation-based degradation process with the prediction of this denoising network.

2) *Decomposition-Based Noise Redefinition*: This kind of methods construct novel diffusion process to recover image \mathbf{x} from its partial measurement \mathbf{c} in the noisy linear inverse problems as follows $\mathbf{c} = \mathbf{H}\mathbf{x} + \mathbf{n}$, where \mathbf{H} is a known linear degradation matrix, $\mathbf{n} \sim \mathcal{N}(0, \sigma_c^2 \mathbf{I})$ is an i.i.d. additive Gaussian noise with known variance. In practice, numerous restoration tasks including inpainting, super-resolution, colorization can be written in form of this noisy linear inverse problems. SVD Decomposition-based methods firstly perform SVD decomposition on the linear degradation matrix \mathbf{H} to decouples the components in the measurement \mathbf{c} . Thereby, the components in measurement \mathbf{c} on spectral space can be viewed as a noisy version of their counterparts derived from clean image \mathbf{x} . In order to incorporate the measurement \mathbf{c} into the diffusion process while preventing the mismatch in noise-level caused by the noise in measurement \mathbf{c} , decomposition-based methods design a proper noise sequence to link the noise in the measurement \mathbf{c} with the noise added in the standard diffusion process. It can be proven that the optimized unconditional denoising network pre-trained on the prior of clean image \mathbf{x} is also the optimal solution for the variational objective of the designed novel diffusion process. Thereby, we can perform sampling process in the spectral space to recover clean image \mathbf{x} from its noisy counterpart \mathbf{c} based on pre-trained unconditional denoising network. SNIPS [138] and DDRM [139] construct SVD decomposition-based novel diffusion process in spectral space based on the annealed Langevin dynamics framework provided by NCSN [194] and the Markov chain diffusion process provided by DDPM [177] respectively.

Different from SNIPS and DDRM, DDNM [144] construct a general solution $\hat{\mathbf{x}}$ based on range-null space decomposition which holds $\mathbf{H}\hat{\mathbf{x}} \equiv \mathbf{c}$. In each denoising step, DDNM [144] project the denoising output $\mathbf{x}_{0|t}$ onto the general solution to guarantee the consistency between denoising output $\mathbf{x}_{0|t}$ and given measurement \mathbf{c} . For noisy linear inverse problem $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}$, DDNM [144] incorporates a scaling factor into the formulation of general solution and designs noise sequence corresponding to the scaling factor during sampling process to assure the noise level in \mathbf{x}_{t-1} aligned with the definition of

$q(\mathbf{x}_{t-1} | \mathbf{x}_0)$ for pre-trained unconditional denoising network.

E. Guidance

In the field of conditional image synthesis, an intuitive idea to sample from the conditional distribution $p(\mathbf{x} | \mathbf{c})$ is approximating the conditional score function $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t | \mathbf{c})$ with conditional denoising network $\epsilon_{\theta}(\mathbf{x}_t, t, \mathbf{c})$. Guidance provides another pathway to approximate the conditional score function without time-consuming conditional training, since the conditional score function can be decomposed into an unconditional score function and the gradient of log likelihood as follows:

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t | \mathbf{c}) = \nabla_{\mathbf{x}_t} \log p_t(\mathbf{c} | \mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) \quad (9)$$

where the score function $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$ can be estimated by an unconditional denoising network $\epsilon_{\theta}(\mathbf{x}_t, t)$. Guidance-based methods design task-specific guidance loss function to reflect the consistency between intermediate latent variable \mathbf{x}_t and conditional inputs \mathbf{c} at each time step t , which serves as the estimation for the log likelihood $\log p_t(\mathbf{c} | \mathbf{x}_t)$.

For multiple conditional inputs, guidance can also be employed to perform conditional control for part of the conditional inputs. In practice, we can split the conditional inputs \mathbf{c} into components \mathbf{c}_0 and \mathbf{c}_1 which are incorporate into the diffusion synthesis framework with conditional denoising network and guidance respectively. In this case, the conditional score function can be written as $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t | \mathbf{c}_0, \mathbf{c}_1) = \nabla_{\mathbf{x}_t} \log p_t(\mathbf{c}_1 | \mathbf{x}_t, \mathbf{c}_0) + \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t | \mathbf{c}_0)$. In this formulation, $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t | \mathbf{c}_0)$ can be estimated by a denoising network conditioned on \mathbf{c}_0 and the log likelihood $\log p_t(\mathbf{c}_1 | \mathbf{x}_t, \mathbf{c}_0)$ can be estimated with the guidance loss.

Currently, guidance-based methods are employed in a wide range of conditional synthesis scenarios with designed task-specific guidance loss. Subsequently, we categorize these approaches based on the target applications.

1) *Classifier Guidance*: The pioneer guidance work Classifier Guidance [145] trains an auxiliary classifier $p_{\phi}(\mathbf{c} | \mathbf{x}_t)$ as the guidance loss function for image synthesis conditioned on class label \mathbf{c} . However, for more complicated conditional control signal \mathbf{c} beyond the class label, training an accurate classifier $p_{\phi}(\mathbf{c} | \mathbf{x}_t)$ is challenging. Therefore, followed up works designs more flexible guidance loss without training or optimizing to handle more complicate tasks.

2) *Guidance for Inverse Problems*: As mentioned in Sec. IV-D2, a wide range of restoration tasks can be expressed by recovering clean image \mathbf{x} from a given partial measurement \mathbf{c} in form of noisy inverse problem: $\mathbf{c} = \mathcal{A}(\mathbf{x}) + \mathbf{n}$, $\mathbf{n} \sim \mathcal{N}(0; \sigma_c^2 \mathbf{I})$, where \mathcal{A} is a known degradation function and \mathbf{n} denotes the additive noise. In practice, approximating the likelihood $p_t(\mathbf{c} | \mathbf{x}_t)$ and perform guidance on sampling process is a widely employed strategy to solve noisy inverse problem. Fig. 7 provides an illustration of sampling process with guidance for inverse problem.

MCG [146] and DPS [147] approximate the gradient of likelihood as follows: $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{c} | \mathbf{x}_t) \approx \nabla_{\mathbf{x}_t} \log p(\mathbf{c} | \mathbf{x}_{0|t}) = -\frac{1}{\sigma_c^2} \nabla_{\mathbf{x}_t} \|\mathbf{c} - \mathcal{A}(\mathbf{x}_{0|t})\|_2^2$. The error of this estimation can be proven to converge to 0 as $\sigma_c \rightarrow \infty$ in most inverse problems.

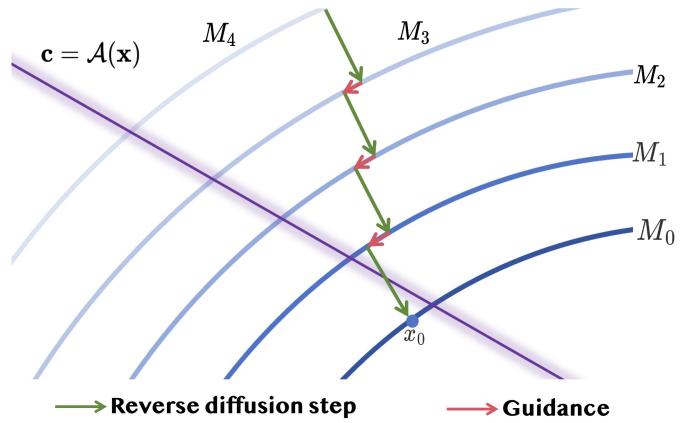


Fig. 7: An illustration of the guided sampling process for inverse problems. The curve M_t denotes the data manifold of intermediate diffuse output \mathbf{x}_t . The guidance process (red arrow) moves \mathbf{x}_t towards the data manifold satisfying the constrain $\mathbf{c} = \mathcal{A}(\mathbf{x})$, which is denoted as the purple line.

IIGDM [153] provides a more accurate estimation for the likelihood by approximating $p_t(\mathbf{x}_0 | \mathbf{x}_t)$ with a Gaussian distribution averaged on $\mathbf{x}_{0|t}$. In order to perform these guidance approaches for inverse problems on diffusion framework on latent space [19], PSLD [154] adds an additional guidance term measuring the reconstruction ability of the intermediate denoising output $\mathbf{z}_{0|t}$ to avoid guiding the sampling trajectory towards latent variable \mathbf{z}_0 away from the manifold of real data.

However, these guidance approaches can only estimate the likelihood term in inverse problems with known concrete form of the degradation operator $\mathcal{A}(\cdot)$. This hinders the deployment of these approaches for unknown real world degradation. BlindDPS [155] explores the applicability of DPS to blind inverse problems, in which degradation operator $\mathcal{A}_{\varphi}(\cdot)$ is parameterized with unknown parameter φ . In order to identify the degradation parameter along with the sampling process for desired image, BlindDPS trains a diffusion model for the parameter φ in degradation operator. In sampling process, BlindDPS employed the similar approximation strategy as DPS [147] to estimate the likelihood term as follows:

$$p_t(\mathbf{c} | \mathbf{x}_t, \varphi_t) \approx p(\mathbf{c} | \mathbf{x}_{0|t}, \varphi_{0|t}). \quad (10)$$

Subsequently, BlindDPS performs parallel sampling process to simultaneously recover the clean image \mathbf{x} and the unknown degradation parameter φ from conditional distribution $p(\mathbf{x}, \varphi | \mathbf{c})$ with the estimated likelihood in Eq.10.

GDP [156] offers a heuristic approximation for the likelihood term, which consists of a distance metric measuring the consistency to conditional inputs and a optional quality enhancement loss to control some desired properties in synthesized results. GDP can also be employed in blind inverse problems by optimizing the degradation parameters in degradation function \mathcal{A} with the distance metric during sampling process.

3) *Guidance for Semantic Control*: Guidance can also be employed to ensure the consistency of diffused output and provided semantic control signals including text prompts or

semantic images without time-consuming fine-tuning or training. In practice, semantic guidance loss is usually designed based on pre-trained CLIP model which learned a rich shared embedding space for image and text.

Blend Diffusion [148] is the pioneer work in the field of semantic guidance, which targets to inpaint the masked region c_m in source image c_I according to the provided text description c_d . Blend Diffusion designs a CLIP guidance loss for the conditional inputs $\mathbf{c} = (c_m, c_I, c_d)$ as follows:

$$L(\mathbf{x}_t, \mathbf{c}) = \mathcal{D}_{CLIP}(\mathbf{x}_{0|t}, \mathbf{c}) + \lambda \mathcal{D}_{bg}(\mathbf{x}_{0|t}, \mathbf{c}), \quad (11)$$

where \mathcal{D}_{CLIP} measures the CLIP distance between the intermediate denoising output $\mathbf{x}_{0|t}$ and text description c_d in mask region for semantic-level alignment, and \mathcal{D}_{bg} calculates the MSE and LPIPS similarity between $\mathbf{x}_{0|t}$ and source image c_I in unmasked region for the faithfulness to source image.

In order to control the sampling process with both provided text prompt and style reference image, SDG [157] employs a linear combination of the CLIP distance from current denoising output to both text embedding and reference image embedding as the guidance loss. DiffuseIT [158] introduce a more comprehensive guidance loss to perform image editing in accordance with given text prompt or style reference image. In addition to the CLIP distance, DiffuseIT also incorporates a structure loss calculated based on the self-attention features of the source image extracted from the Vision Transformer (ViT) to better preserve the structure of the source image.

4) *Guidance for Visual Signals*: In practice, a branch of works employ guidance to control the consistency between diffuse output and given visual signal. In order to measure the consistency between intermediate diffuse output and provided visual signal, some works train neural networks to project the intermediate diffuse output \mathbf{x}_t onto its corresponding visual signal and leverage distance metric as the guidance loss for sketch-to-image [149] and stroke-to-image [159]. Readout Guidance [160] provide a unified guidance-based framework for diverse visual signal to image task by training various readout heads to synthesize different task-specific visual feature maps reflecting the spatial layout or inherent correspondence in images to perform guidance. Different from these works, FreeControl [161] prefers to impose guidance loss on the difference in the space of PCA components of self-attention map between the intermediate diffuse output and visual signal.

5) *Guidance for Attention Layers*: In DM-based conditional image synthesis, the attention layers in denoising network effectively control the layout, structure and semantics of synthesized image. However, directly manipulating the attention layers through replacement or localization as described in Section IV-B introduces artificial modifications to the internal parameters of the denoising network and may impair its modeling capability. Therefore, a branch of works employ guidance to achieve softly control for attention layers.

For image editing, attention guidance is performed as substitution of attention replacement to softly control the consistency between source image and edited result. Pix2Pix-Zero [150] employs a guidance loss measuring the L_2 distance between the cross-attention maps in editing branch and reconstruction

branch instead of the replacement manipulation in Prompt-to-prompt [123]. In order to find a more expressive attention map as guide reference, Rediffuser [162] employs a sliding fusion strategy to fuse the cross-attention maps obtained from sampling branches conditioned on source prompt, target prompt and an intermediate representation. EBMs [163] employs a energy function to guide the integration of the semantic information in editorial prompts with the structure and layout of source image restored in cross-attention layers.

Attention guidance can also be employed to perform attention localization. For object-level layout control, Chen et.al [164] employs guidance to control the cross-attention map, which locates the objects in text prompts into their desired bounding boxes. Self-guidance [165] extracts the various characteristics including position, size, shape and appearance of the desired object from the intermediate activations and attention maps. Subsequently, Self-guidance places constraints on these characteristics with guidance loss measuring their consistency to desired conditional control signal. For drag-based editing tasks which target to move certain foreground contents in source image into target region, Dragondiffusion [67] designs energy functions based on the cosine distance between intermediate features in the U-Net decoder as guidance to ensure correspondence between the original content region and target dragging region. DiffEditor [166] develops the guidance framework of DragonDiffusion [67] by introducing SDE-based sampling process on the masked region instead of ODEs to improve editing flexibility.

6) *Enhanced Guidance framework*: In some complicated conditional synthesis scenarios, simply incorporating the gradient of guidance loss in each denoising step may lead to artifacts and strange behaviors because of the failure in balancing the realness and guidance constraint satisfaction in guided sampling process. Therefore, some state-of-the-art guidance works provide enhanced unified guidance frameworks to more effectively fuse the prior knowledge in pre-trained model and the information in control signals. FreeDoM [151] employs a time-travel strategy that rolls back the intermediate latent variable \mathbf{x}_t to a certain previous time step \mathbf{x}_{t+j} and resamples it to time step t again. This strategy inserts additional steps into the guided sampling process, allowing for a more seamless integration of the information from the pre-trained model and the conditional control signals. In order to enhance the consistency to conditional control signals, Universal Guidance [152] performs an m -step gradient descent optimization process to find the point with minimum guidance loss in the vicinity of the intermediate denoising output $\mathbf{x}_{0|t}$. Subsequently, this point is employed to infer the next latent variable \mathbf{x}_{t-1} .

F. Conditional Correction

In some conditional synthesis scenarios, the synthesized images are controlled by the constraints specified by conditional inputs \mathbf{c} (such as the formulation of inverse problems). To ensure the synthesized result to be consistent to the inputs \mathbf{c} , conditional correction-based methods perform a correction operator on the intermediate diffuse output \mathbf{x}_t (or $\mathbf{x}_{0|t}$), which directly projects the current diffuse output onto the data

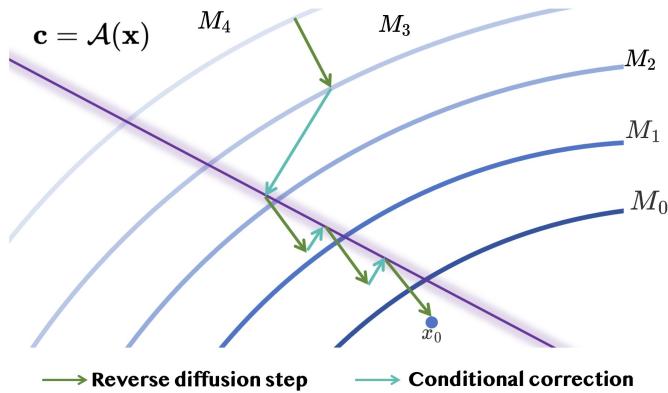


Fig. 8: An illustration of the sampling process with conditional correction for inverse problem. The conditional correction process (cyan arrow) projects \mathbf{x}_t onto the data manifold satisfying the constrain $\mathbf{c} = \mathcal{A}(\mathbf{x})$.

manifold satisfying the constrain imposed by given conditional control signal \mathbf{c} . Subsequently, this corrected latent variable will be pass into next denoising step. Fig. 8 provides an illustration of sampling process with conditional correction for inverse problem.

Currently, conditional correction are widely employed in image inpainting tasks, which involves synthesizing content for the masked region \mathbf{c}_m in incomplete reference image \mathbf{c}_y . The constrain in inpainting tasks can be expressed as: $\mathbf{c}_y = (1 - \mathbf{c}_m) \odot \mathbf{x}$. Pioneer diffusion work Song et al. [167] performs inpainting based on conditional correction by replacing the unmask region in denoising output $\mathbf{x}_{0|t}$ with its counterpart in reference image \mathbf{c}_y to ensure the faithfulness to the content in unmasked region.

Different from Song et al. [167], Repaint [168] prefers to perform replacement correction on latent variable \mathbf{x}_t . Besides, Repaint rolls back the intermediate latent variable \mathbf{x}_t to the previous time step and resamples it to time step t several times to eliminate the artifacts caused by conditional correction. The constrain in Super-resolution task can be written as: $\mathbf{c} = \phi_N \mathbf{x}$, where \mathbf{c} denotes the low-resolution image of \mathbf{x} downsampled by degradation matrix ϕ_N with factor N . ILVR [169] performs conditional correction by substituting the low-frequency components in latent variable with its counterpart noisy low-resolution image to the consistency between degraded latent variable and its counterpart noisy reference low-resolution image.

Conditional correction are also widely employed in image editing tasks to preserve the background not requiring editing [170, 172–175]. With the provided mask for background in source image, text-based image editing tasks can be viewed as performing image inpainting for the foreground region based on given text prompt. However, the provided mask for background is always not available in editing tasks. Therefore, a branch of works propose approaches to generate masks or segmentation automatically by inferring the reasonable layout for the user-desired edited image based on the given source image and text prompt. Diffedit [170] identifies the mask for background by comparing differences in the denoising

outputs of noisy source image conditioned on source prompt and target prompt. Object-Shape Variation [172] segments the provided source image by aggregating the attention map into clusters corresponding to different semantic segments and identifying the segments with the nouns in the text prompt based on the similarity between the segments and the cross-attention map of noun tokens. Besides, a branch of works [173–175] employ pre-trained image segmentation modules to automatically generate masks or segmentation according to the structure information in the given source image and text prompt.

CCDF [171] proposes a general conditional correction formula for constrains in form of general noisy linear inverse problem. In practice, the conditional correction operator in [167–169] can be expressed in the general form provided by CCDF. Besides, CCDF provides a theoretical basis for the faithfulness of this corrected sampling trajectory to original sampling process. CCDF proves when the linear degradation operator \mathbf{H} is a non-expansive mapping, the upper bound of the deviation in final output \mathbf{x}_0 will converge to a constant as the total diffusion step $T \rightarrow \infty$. MCG [146] further performs guidance on conditional correction framework provided by CCDF, which alleviates the deviation from original sampling process caused by conditional correction.

V. CHALLENGES AND FUTURE DIRECTIONS

Although DM-based conditional image synthesis has made remarkable progress in generating high-quality images aligned with various user-provided conditions, there remains a significant disparity between academic advancements and practical needs for conditional image synthesis. In this section, we summarize several main challenges in this field and identify potential solutions to address them in the future.

A. Sampling Acceleration

The time-consuming sampling process often creates a bottleneck of diffusion-based image synthesis, and its acceleration will facilitate the model deployment in practice [235, 236]. Early works on sampling acceleration are devoted to reducing the number of sampling steps with better numerical solvers [179, 187, 233, 234, 237] or distilling the sampling process of pre-trained diffusion models to build short-cuts that enable faster sampling [189, 238–240]. However, too few denoising steps with the distilled model may compromise the effectiveness of in-sampling condition integration.

An important type of current sampling acceleration works reduces the computational cost of each denoising step by decreasing model parameters using techniques such as knowledge distillation [241, 242] and architecture search [235, 236, 243]. Most of DM-based parameter compression approaches are currently tailored for text-to-image models. Analyzing whether the parameter redundancy also exists for models of other conditional synthesis tasks, similar to those in text-to-image models, and extending these model compression methods to more complicated downstream tasks, is another promising future direction.

B. Artifacts Caused by In-sampling Conditioning Mechanisms

In-sampling condition mechanisms summarized in Sec. IV allows for flexible condition integration in DM-based image synthesis without performing time-consuming condition integration for the denoising network. However, these conditioning mechanisms introduce modification to the standard sampling process in diffusion framework and lead to deviations from the modeled data distribution, which resulting in artifacts in synthesized images [150–152, 168]. The vast majority of works resort to complex adjustment mechanisms to address the artifact issue caused by in-sampling condition integration. This includes time-step rolling back for guidance [151], localization for attention map [117, 125] and diffusion process revision for restoration tasks [137, 139]. However, these methods are highly customized based on specific application scenarios. A feasible future direction for developing more generic solution is to perform lightweight fine-tuning on the denoising network with the diffusion loss based on the intermediate latent variables in the sampling process equipped with in-sampling conditioning mechanisms. This tends to smooth out artifacts under in-sampling conditioning mechanisms and synthesize desire images in a lower computational cost comparing to perform condition integration in denoising network .

C. Training Datasets

Among the various conditioning mechanisms, the most fundamental and effective pathway for condition integration is still the supervised learning on pairs of conditional input and image. Although training datasets are relatively sufficient for conditional synthesis tasks involving single modality conditional inputs, such as text-to-image [244, 245], restoration [246–248], and visual signal to image [249–251], gathering enough data for tasks with complex, multi-modal conditional inputs like image editing, customization, and composition remains challenging. With the advancement of training and efficient fine-tuning techniques for large language models, various types of large models are constantly being developed with powerful multi-modal representation learning [223, 226, 227] and content generation abilities [123, 124], making it possible to leverage these pre-trained models to automatically produce desired training datasets. We may also consider self-supervised or weakly supervised learning to reduce the demand for a large amount of high-quality training data [50, 83, 87].

D. Robustness

Due to the lack of objective task-specific evaluation datasets and metrics in some complex tasks, studies for these tasks prefer to compare models based on a set of self-defined conditional inputs, making the performance appear overly optimistic. In fact, many renowned text-to-image models [19, 20, 22] have been found to produce unsatisfactory synthesized results for certain specific categories of text prompts, as demonstrated by the shortcomings of Imagen [20] in generating facial images.

Here we point out some pathways to address issues of robustness. First, for conditional inputs where the model

performs poorly, augmenting the training dataset is a direct approach. Second, the difficulties to handle conditional inputs in a certain category may be due to the insufficient capability or unsuitability of the conditional encoder with this category of data. In this case, incorporating encoder architectures tailored for this data category into the conditional encoder, or designing more capable compound conditional encoders, becomes a preferable choice. Besides, performing specialization for given conditional inputs is also an effective pathway to provide robust results at the cost of time-consuming fine-tuning or optimization. Finally, employ sampling process conditioning mechanisms, such as guidance, conditional correction and attention manipulation, to achieve more detailed control can also prevent undesired synthesis results.

E. Safety

The developments in AI-generated content (AIGC) propelled by the superior performance of diffusion-based conditional synthesis and their downstream applications lead to severe safety concerns in aspects of bias and fairness, copyright, and the risk of exposure to harmful content. Safety-oriented DM-based conditional image synthesis is dedicated to mitigating these issues by embedding watermarks that are easily reproducible in DM-generated images to detect copyright infringement [252–254], and reducing bias by increasing model's orientation towards minority groups in basic unconditional or text-conditioned synthesis via classic conditioning mechanisms, such as fine-tuning [255], guidance [256], and conditional correction [257]. Efforts have also been made in preventing harmful contents in the text-to-image task via harmful prompt detection [19], prompt engineering [257] and safety guidance [258]. The current safety-focused efforts mainly concentrate on basic unconditional or text-conditioned synthesis. We believe that for more complex conditional synthesis scenarios, safety-oriented efforts in this area can be focused on four main aspects: (a) detecting harmful conditional inputs, (b) filtering and removing bias from the training dataset, (c) providing safety-focused guidance for the sampling process, and (d) implementing safety-focused fine-tuning of the denoising network.

VI. CONCLUSION

This survey presents a thorough investigation of DM-based conditional image synthesis, focusing on framework-level construction and common design choices behind various conditional image synthesis problems across seven representative categories of tasks. Despite the progress made, efforts are still needed in the future to handle challenges in practical applications. Future researches should focus on gathering and creating sufficient high-quality and unbiased task-specific datasets, carefully designed conditional encoder architectures and in-sampling conditioning mechanisms for effective and robust conditional modeling to synthesize stable and flawless results. Trade-off between fast sampling and synthesization quality and is also a key issue for practical deployment. Finally, as a popular AIGC technology, it is necessary to fully consider the safety issues and legitimacy it brings.

REFERENCES

- [1] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, “Generative adversarial text to image synthesis,” in *ICML*, 2016.
- [2] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas, “Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks,” in *ICCV*, 2017.
- [3] M. Ding, Z. Yang, W. Hong, W. Zheng, C. Zhou, D. Yin, J. Lin, X. Zou, Z. Shao, H. Yang *et al.*, “Cogview: Mastering text-to-image generation via transformers,” in *NeurIPS*, 2021.
- [4] P. Esser, R. Rombach, and B. Ommer, “Taming transformers for high-resolution image synthesis,” in *CVPR*, 2021.
- [5] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, “Zero-shot text-to-image generation,” in *ICML*, 2021.
- [6] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, “Photo-realistic single image super-resolution using a generative adversarial network,” in *CVPR*, 2017.
- [7] X. Wang, L. Xie, C. Dong, and Y. Shan, “Real-esrgan: Training real-world blind super-resolution with pure synthetic data,” in *ICCV*, 2021, pp. 1905–1914.
- [8] L. Maaløe, M. Fraccaro, V. Liévin, and O. Winther, “Biva: A very deep hierarchy of latent variables for generative modeling,” in *NeurIPS*, 2019.
- [9] D. Lee, C. Kim, S. Kim, M. Cho, and W.-S. Han, “Autoregressive image generation using residual quantization,” in *CVPR*, 2022.
- [10] A. Brock, T. Lim, J. M. Ritchie, and N. Weston, “Neural photo editing with introspective adversarial networks,” in *ICLR*, 2017.
- [11] H. Ling, K. Kreis, D. Li, S. W. Kim, A. Torralba, and S. Fidler, “Editgan: High-precision semantic image editing,” in *NeurIPS*, 2021.
- [12] R. Abdal, Y. Qin, and P. Wonka, “Image2stylegan++: How to edit the embedded images?” in *CVPR*, 2020.
- [13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *NeurIPS*, 2014.
- [14] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *arXiv:1411.1784*, 2014.
- [15] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” in *ICLR*, 2014.
- [16] K. Sohn, H. Lee, and X. Yan, “Learning structured output representation using deep conditional generative models,” in *NeurIPS*, 2015.
- [17] A. Van Den Oord, N. Kalchbrenner, and K. Kavukcuoglu, “Pixel recurrent neural networks,” in *ICML*, 2016.
- [18] A. Van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves *et al.*, “Conditional image generation with pixelcnn decoders,” in *NeurIPS*, 2016.
- [19] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *CVPR*, 2022.
- [20] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans *et al.*, “Photorealistic text-to-image diffusion models with deep language understanding,” in *NeurIPS*, 2022.
- [21] A. Q. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen, “GLIDE: towards photorealistic image generation and editing with text-guided diffusion models,” in *ICML*, 2022.
- [22] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, “Hierarchical text-conditional image generation with CLIP latents,” *arXiv:2204.06125*, 2022.
- [23] Y. Balaji, S. Nah, X. Huang, A. Vahdat, J. Song, K. Kreis, M. Aittala, T. Aila, S. Laine, B. Catanzaro *et al.*, “ediffi: Text-to-image diffusion models with an ensemble of expert denoisers,” *arXiv:2211.01324*, 2022.
- [24] S. Gu, D. Chen, J. Bao, F. Wen, B. Zhang, D. Chen, L. Yuan, and B. Guo, “Vector quantized diffusion model for text-to-image synthesis,” in *CVPR*, 2022.
- [25] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi, “Image super-resolution via iterative refinement,” *IEEE TPAMI*, 2022.
- [26] J. Ho, C. Saharia, W. Chan, D. J. Fleet, M. Norouzi, and T. Salimans, “Cascaded diffusion models for high fidelity image generation,” *JMLR*, 2022.
- [27] C. Saharia, W. Chan, H. Chang, C. Lee, J. Ho, T. Salimans, D. Fleet, and M. Norouzi, “Palette: Image-to-image diffusion models,” in *ACM SIGGRAPH*, 2022.
- [28] H. Li, Y. Yang, M. Chang, S. Chen, H. Feng, Z. Xu, Q. Li, and Y. Chen, “Srdiff: Single image super-resolution with diffusion probabilistic models,” *Neurocomputing*, 2022.
- [29] H. Sahak, D. Watson, C. Saharia, and D. Fleet, “Denoising diffusion probabilistic models for robust image super-resolution in the wild,” *arXiv:2302.07864*, 2023.
- [30] S. Shang, Z. Shan, G. Liu, L. Wang, X. Wang, Z. Zhang, and J. Zhang, “Resdiff: Combining cnn and diffusion model for image super-resolution,” in *AAAI*, 2024.
- [31] Y. Zhang, J. Zhang, H. Li, Z. Wang, L. Hou, D. Zou, and L. Bian, “Diffusion-based blind text image super-resolution,” in *CVPR*, 2024.
- [32] H. Jiang, A. Luo, H. Fan, S. Han, and S. Liu, “Low-light image enhancement with wavelet-based diffusion models,” *ACM TOG*, 2023.
- [33] M. Xue, J. He, Y. He, Z. Liu, W. Wang, and M. Zhou, “Low-light image enhancement via clip-fourier guided wavelet diffusion,” *arXiv:2401.03788*, 2024.
- [34] C. Zhao, W. Cai, C. Dong, and C. Hu, “Wavelet-based fourier information interaction with frequency diffusion adjustment for underwater image restoration,” in *CVPR*, 2024.
- [35] K. Preechakul, N. Chathee, S. Wizadwongsu, and S. Suwanjanakorn, “Diffusion autoencoders: Toward a meaningful and decodable representation,” in *CVPR*, 2022.
- [36] W. Wang, J. Bao, W. Zhou, D. Chen, D. Chen, L. Yuan, and H. Li, “Semantic image synthesis via diffusion models,” *arXiv:2207.00050*, 2022.
- [37] K. Zhang, M. Sun, J. Sun, B. Zhao, K. Zhang, Z. Sun, and T. Tan, “Humandiffusion: a coarse-to-fine alignment diffusion framework for controllable text-driven person image generation,” *arXiv:2211.06235*, 2022.
- [38] Y. Li, H.-C. Shao, X. Liang, L. Chen, R. Li, S. Jiang, J. Wang, and Y. Zhang, “Zero-shot medical image translation via frequency-guided diffusion models,” *IEEE Trans. Med. Imaging*, 2023.
- [39] J. Liu, R. Anirudh, J. J. Thiagarajan, S. He, K. A. Mohan, U. S. Kamilov, and H. Kim, “Dolce: A model-based probabilistic diffusion framework for limited-angle ct reconstruction,” in *ICCV*, 2023.
- [40] P. A. Moghadam, S. Van Dalen, K. C. Martin, J. Lennerz, S. Yip, H. Farahani, and A. Bashashati, “A morphology focused diffusion probabilistic model for synthesis of histopathology images,” in *WACV*, 2023.
- [41] X. Meng, Y. Gu, Y. Pan, N. Wang, P. Xue, M. Lu, X. He, Y. Zhan, and D. Shen, “A novel unified conditional score-based generative framework for multi-modal medical image completion,” *arXiv:2207.03430*, 2022.
- [42] L. Yang, Z. Huang, Y. Song, S. Hong, G. Li, W. Zhang, B. Cui, B. Ghanem, and M.-H. Yang, “Diffusion-based scene graph to image generation with masked contrastive pre-training,” *arXiv:2211.11138*, 2022.
- [43] A. Graikos, S. Yellapragada, M.-Q. Le, S. Kapse, P. Prasanna, J. Saltz, and D. Samaras, “Learned representation-guided diffusion models for large-image generation,” *arXiv:2312.07330*, 2023.

- [44] C. Mou, X. Wang, L. Xie, Y. Wu, J. Zhang, Z. Qi, and Y. Shan, “T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models,” in *AAAI*, 2024.
- [45] L. Zhang, A. Rao, and M. Agrawala, “Adding conditional control to text-to-image diffusion models,” in *ICCV*, 2023.
- [46] T. Wang, T. Zhang, B. Zhang, H. Ouyang, D. Chen, Q. Chen, and F. Wen, “Pretraining is all you need for image-to-image translation,” *arXiv:2205.12952*, 2022.
- [47] D. Li, J. Li, and S. C. Hoi, “Blip-diffusion: pre-trained subject representation for controllable text-to-image generation and editing,” in *NeurIPS*, 2023.
- [48] T.-J. Fu, W. Hu, X. Du, W. Y. Wang, Y. Yang, and Z. Gan, “Guiding instruction-based image editing via multimodal large language models,” *arXiv:2309.17102*, 2023.
- [49] P. Kocsis, J. Philip, K. Sunkavalli, M. Nießner, and Y. Hold-Geoffroy, “Lightit: Illumination modeling and control for diffusion models,” in *CVPR*, 2024.
- [50] X. Zhang, J. Guo, P. Yoo, Y. Matsuo, and Y. Iwasawa, “Paste, inpaint and harmonize via denoising: Subject-driven image editing with pre-trained diffusion model,” *arXiv:2306.07596*, 2023.
- [51] V. Goel, E. Peruzzo, Y. Jiang, D. Xu, N. Sebe, T. Darrell, Z. Wang, and H. Shi, “Pair-diffusion: Object-level image editing with structure-and-appearance paired diffusion models,” *arXiv:2303.17546*, 2023.
- [52] Y. Yang, H. Peng, Y. Shen, Y. Yang, H. Hu, L. Qiu, H. Koike *et al.*, “Imagebrush: Learning visual in-context instructions for exemplar-based image manipulation,” in *NeurIPS*, 2024.
- [53] X. Xu, J. Guo, Z. Wang, G. Huang, I. Essa, and H. Shi, “Prompt-free diffusion: Taking “text” out of text-to-image diffusion models,” in *CVPR*, 2024.
- [54] G. Xiao, T. Yin, W. T. Freeman, F. Durand, and S. Han, “Fastcomposer: Tuning-free multi-subject image generation with localized attention,” *arXiv:2305.10431*, 2023.
- [55] J. Ma, J. Liang, C. Chen, and H. Lu, “Subject-diffusion: Open domain personalized text-to-image generation without test-time fine-tuning,” in *ACM SIGGRAPH*, 2024.
- [56] R. Gal, M. Arar, Y. Atzmon, A. H. Bermano, G. Chechik, and D. Cohen-Or, “Encoder-based domain tuning for fast personalization of text-to-image models,” *ACM TOG*, 2023.
- [57] X. Jia, Y. Zhao, K. C. Chan, Y. Li, H. Zhang, B. Gong, T. Hou, H. Wang, and Y.-C. Su, “Taming encoder for zero fine-tuning image customization with text-to-image diffusion models,” *arXiv:2304.02642*, 2023.
- [58] X. Li, M. Kampffmeyer, X. Dong, Z. Xie, F. Zhu, H. Dong, X. Liang *et al.*, “Warpdiffusion: Efficient diffusion model for high-fidelity virtual try-on,” *arXiv:2312.03667*, 2023.
- [59] Y. Lu, M. Zhang, A. J. Ma, X. Xie, and J. Lai, “Coarse-to-fine latent diffusion for pose-guided person image synthesis,” in *CVPR*, 2024.
- [60] J. Shi, W. Xiong, Z. Lin, and H. J. Jung, “Instantbooth: Personalized text-to-image generation without test-time finetuning,” in *CVPR*, 2024.
- [61] K. Shiohara and T. Yamasaki, “Face2diffusion for fast and editable face personalization,” in *CVPR*, 2024.
- [62] Y. Feng, B. Gong, D. Chen, Y. Shen, Y. Liu, and J. Zhou, “Ranni: Taming text-to-image diffusion for accurate instruction following,” *arXiv:2311.17002*, 2023.
- [63] Y. Huang, L. Xie, X. Wang, Z. Yuan, X. Cun, Y. Ge, J. Zhou, C. Dong, R. Huang, R. Zhang *et al.*, “Smartedit: Exploring complex instruction-based image editing with multimodal large language models,” *arXiv:2312.06739*, 2023.
- [64] S. Li, H. Singh, and A. Grover, “Instructany2pix: Flexible visual editing via multimodal instruction following,” *arXiv:2312.06738*, 2023.
- [65] H. Ye, J. Zhang, S. Liu, X. Han, and W. Yang, “Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models,” *arXiv:2308.06721*, 2023.
- [66] Y. Li, H. Liu, Q. Wu, F. Mu, J. Yang, J. Gao, C. Li, and Y. J. Lee, “GLIGEN: open-set grounded text-to-image generation,” in *CVPR*, 2023.
- [67] C. Mou, X. Wang, J. Song, Y. Shan, and J. Zhang, “Dragondiffusion: Enabling drag-style manipulation on diffusion models,” in *ICLR*, 2024.
- [68] Y. Wei, Y. Zhang, Z. Ji, J. Bai, L. Zhang, and W. Zuo, “Elite: Encoding visual concepts into textual embeddings for customized text-to-image generation,” in *CVPR*, 2023.
- [69] J. T. Hoe, X. Jiang, C. S. Chan, Y.-P. Tan, and W. Hu, “Interactdiffusion: Interaction control in text-to-image diffusion models,” *arXiv:2312.05849*, 2023.
- [70] X. Wang, T. Darrell, S. S. Rambhatla, R. Girdhar, and I. Misra, “Instancediffusion: Instance-level control for image generation,” in *CVPR*, 2024.
- [71] T. Qi, S. Fang, Y. Wu, H. Xie, J. Liu, L. Chen, Q. He, and Y. Zhang, “Deadiff: An efficient stylization diffusion model with disentangled representations,” in *CVPR*, 2024.
- [72] Y. Gu, X. Wang, J. Z. Wu, Y. Shi, Y. Chen, Z. Fan, W. Xiao, R. Zhao, S. Chang, W. Wu *et al.*, “Mix-of-show: Decentralized low-rank adaptation for multi-concept customization of diffusion models,” in *NeurIPS*, 2024.
- [73] T. Brooks, A. Holynski, and A. A. Efros, “Instructpix2pix: Learning to follow image editing instructions,” in *CVPR*, 2023.
- [74] B. Yang, S. Gu, B. Zhang, T. Zhang, X. Chen, X. Sun, D. Chen, and F. Wen, “Paint by example: Exemplar-based image editing with diffusion models,” in *CVPR*, 2023.
- [75] A. B. Yildirim, V. Baday, E. Erdem, A. Erdem, and A. Dundar, “Inst-inpaint: Instructing to remove objects with diffusion models,” *arXiv:2304.03246*, 2023.
- [76] J. Wei, S. Wu, X. Jiang, and Y. Wang, “Dialogpaint: A dialog-based image editing model,” *arXiv:2303.10073*, 2023.
- [77] K. Zhang, L. Mo, W. Chen, H. Sun, and Y. Su, “Magicbrush: A manually annotated dataset for instruction-guided image editing,” in *NeurIPS*, 2024.
- [78] Z. Geng, B. Yang, T. Hang, C. Li, S. Gu, T. Zhang, J. Bao, Z. Zhang, H. Hu, D. Chen *et al.*, “Instructdiffusion: A generalist modeling interface for vision tasks,” *arXiv:2309.03895*, 2023.
- [79] S. Sheynin, A. Polyak, U. Singer, Y. Kirstain, A. Zohar, O. Ashual, D. Parikh, and Y. Taigman, “Emu edit: Precise image editing via recognition and generation tasks,” in *CVPR*, 2024.
- [80] S. Zhang, X. Yang, Y. Feng, C. Qin, C.-C. Chen, N. Yu, Z. Chen, H. Wang, S. Savarese, S. Ermon *et al.*, “Hive: Harnessing human feedback for instructional visual editing,” in *CVPR*, 2024.
- [81] S. Wang, C. Saharia, C. Montgomery, J. Pont-Tuset, S. Noy, S. Pellegrini, Y. Onoe, S. Laszlo, D. J. Fleet, R. Soricut *et al.*, “Imagen editor and editbench: Advancing and evaluating text-guided image inpainting,” in *CVPR*, 2023.
- [82] S. Xie, Z. Zhang, Z. Lin, T. Hinz, and K. Zhang, “Smartbrush: Text and shape guided object inpainting with diffusion model,” in *CVPR*, 2023.
- [83] S. Xie, Y. Zhao, Z. Xiao, K. C. Chan, Y. Li, Y. Xu, K. Zhang, and T. Hou, “Dreaminpainter: Text-guided subject-driven image inpainting with diffusion models,” *arXiv:2312.03771*, 2023.
- [84] Y. Song, Z. Zhang, Z. Lin, S. Cohen, B. Price, J. Zhang, S. Y. Kim, and D. Aliaga, “Objectstitch: Object compositing with diffusion model,” in *CVPR*, 2023.
- [85] K. Kim, S. Park, J. Lee, and J. Choo, “Reference-based image composition with sketch via structure-aware diffusion model,” *arXiv:2304.09748*, 2023.
- [86] X. Chen, L. Huang, Y. Liu, Y. Shen, D. Zhao, and H. Zhao, “Anydoor: Zero-shot object-level image customization,” in *CVPR*, 2024.
- [87] Z. Zhang, J. Zheng, Z. Fang, and B. A. Plummer, “Text-to-image editing by image information removal,” in *WACV*, 2024.
- [88] B. Kawar, S. Zada, O. Lang, O. Tov, H. Chang, T. Dekel,

- I. Mosseri, and M. Irani, “Imagic: Text-based real image editing with diffusion models,” in *CVPR*, 2023.
- [89] R. Gal, Y. Alaluf, Y. Atzmon, O. Patashnik, A. H. Bermano, G. Chechik, and D. Cohen-Or, “An image is worth one word: Personalizing text-to-image generation using textual inversion,” in *ICLR*, 2023.
- [90] Q. Wu, Y. Liu, H. Zhao, A. Kale, T. Bui, T. Yu, Z. Lin, Y. Zhang, and S. Chang, “Uncovering the disentanglement capability in text-to-image diffusion models,” in *CVPR*, 2023.
- [91] S. Mahajan, T. Rahman, K. M. Yi, and L. Sigal, “Prompting hard or hardly prompting: Prompt inversion for text-to-image diffusion models,” in *CVPR*, 2024.
- [92] H. Ravi, S. Kelkar, M. Harikumar, and A. Kale, “Predictor: Text guided image editing with diffusion prior,” *arXiv:2302.07979*, 2023.
- [93] S. Zhang, S. Xiao, and W. Huang, “Forgedit: Text guided image editing via learning and forgetting,” *arXiv:2309.10556*, 2023.
- [94] R. Bodur, E. Gundogdu, B. Bhattacharai, T.-K. Kim, M. Donoser, and L. Bazzani, “iedit: Localised text-guided image editing with weak supervision,” in *CVPR*, 2024.
- [95] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, “Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation,” in *CVPR*, 2023.
- [96] D. Valevski, M. Kalman, E. Molad, E. Segalis, Y. Matias, and Y. Leviathan, “Unitune: Text-driven image editing by fine tuning a diffusion model on a single image,” *ACM TOG*, 2023.
- [97] P. Li, Q. Huang, Y. Ding, and Z. Li, “Layerdiffusion: Layered controlled image editing with diffusion models,” in *SIGGRAPH Asia*, 2023.
- [98] Z. Zhang, L. Han, A. Ghosh, D. N. Metaxas, and J. Ren, “Sine: Single image editing with text-to-image diffusion models,” in *CVPR*, 2023.
- [99] N. Kumari, B. Zhang, R. Zhang, E. Shechtman, and J. Zhu, “Multi-concept customization of text-to-image diffusion,” in *CVPR*, 2023.
- [100] J. Choi, Y. Choi, Y. Kim, J. Kim, and S. Yoon, “Custom-edit: Text-guided image editing with customized diffusion models,” *arXiv:2305.15779*, 2023.
- [101] Z. Liu, R. Feng, K. Zhu, Y. Zhang, K. Zheng, Y. Liu, D. Zhao, J. Zhou, and Y. Cao, “Cones: concept neurons in diffusion models for customized generation,” in *ICML*, 2023.
- [102] Z. Liu, Y. Zhang, Y. Shen, K. Zheng, K. Zhu, R. Feng, Y. Liu, D. Zhao, J. Zhou, and Y. Cao, “Cones 2: Customizable image synthesis with multiple subjects,” in *NeurIPS*, 2023.
- [103] L. Han, Y. Li, H. Zhang, P. Milanfar, D. Metaxas, and F. Yang, “Svdiff: Compact parameter space for diffusion fine-tuning,” in *ICCV*, 2023.
- [104] C. Meng, Y. He, Y. Song, J. Song, J. Wu, J. Zhu, and S. Ermon, “Sedit: Guided image synthesis and editing with stochastic differential equations,” in *ICLR*, 2022.
- [105] X. Su, J. Song, C. Meng, and S. Ermon, “Dual diffusion implicit bridges for image-to-image translation,” in *ICLR*, 2023.
- [106] R. Mokady, A. Hertz, K. Aberman, Y. Pritch, and D. Cohen-Or, “Null-text inversion for editing real images using guided diffusion models,” in *CVPR*, 2023.
- [107] C. H. Wu and F. De la Torre, “A latent space of stochastic diffusion models for zero-shot image editing and guidance,” in *ICCV*, 2023.
- [108] I. Huberman-Spiegelglas, V. Kulikov, and T. Michaeli, “An edit friendly ddpm noise space: Inversion and manipulations,” *arXiv:2304.06140*, 2023.
- [109] W. Dong, S. Xue, X. Duan, and S. Han, “Prompt tuning inversion for text-driven image editing using diffusion models,” in *ICCV*, 2023.
- [110] J. Huang, Y. Liu, J. Qin, and S. Chen, “Kv inversion: Kv embeddings learning for text-conditioned real image action editing,” in *PRCV*, 2023.
- [111] Z. Wang, L. Zhao, and W. Xing, “Stylediffusion: Controllable disentangled style transfer via diffusion models,” in *ICCV*, 2023.
- [112] B. Wallace, A. Gokul, and N. Naik, “Edict: Exact diffusion inversion via coupled transformations,” in *CVPR*, 2023.
- [113] D. Miyake, A. Iohara, Y. Saito, and T. Tanaka, “Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models,” *arXiv:2305.16807*, 2023.
- [114] X. Ju, A. Zeng, Y. Bian, S. Liu, and Q. Xu, “Direct inversion: Boosting diffusion-based editing with 3 lines of code,” *arXiv:2310.01506*, 2023.
- [115] B. Meiri, D. Samuel, N. Darshan, G. Chechik, S. Avidan, and R. Ben-Ari, “Fixed-point inversion for text-to-image diffusion models,” *arXiv:2312.12540*, 2023.
- [116] Z. Pan, R. Gherardi, X. Xie, and S. Huang, “Effective real image editing with accelerated iterative diffusion inversion,” in *ICCV*, 2023.
- [117] S. Lu, Y. Liu, and A. W.-K. Kong, “Tf-icon: Diffusion-based training-free cross-domain image composition,” in *ICCV*, 2023.
- [118] M. Brack, F. Friedrich, K. Kornmeier, L. Tsaban, P. Schramowski, K. Kersting, and A. Passos, “Ledit++: Limitless image editing using text-to-image models,” in *CVPR*, 2024.
- [119] S. Nie, H. A. Guo, C. Lu, Y. Zhou, C. Zheng, and C. Li, “The blessing of randomness: Sde beats ode in general diffusion-based image editing,” in *ICLR*, 2023.
- [120] Y. Zhang, N. Huang, F. Tang, H. Huang, C. Ma, W. Dong, and C. Xu, “Inversion-based style transfer with diffusion models,” in *CVPR*, 2023.
- [121] J. Chung, S. Hyun, and J.-P. Heo, “Style injection in diffusion: A training-free approach for adapting large-scale diffusion models for style transfer,” in *CVPR*, 2024.
- [122] Y. Shi, C. Xue, J. H. Liew, J. Pan, H. Yan, W. Zhang, V. Y. Tan, and S. Bai, “Dragdiffusion: Harnessing diffusion models for interactive point-based image editing,” in *CVPR*, 2024.
- [123] A. Hertz, R. Mokady, J. Tenenbaum, K. Aberman, Y. Pritch, and D. Cohen-Or, “Prompt-to-prompt image editing with cross-attention control,” in *ICLR*, 2023.
- [124] N. Tumanyan, M. Geyer, S. Bagon, and T. Dekel, “Plug-and-play diffusion features for text-driven image-to-image translation,” in *CVPR*, 2023.
- [125] M. Cao, X. Wang, Z. Qi, Y. Shan, X. Qie, and Y. Zheng, “Masctril: Tuning-free mutual self-attention control for consistent image synthesis and editing,” in *ICCV*, 2023.
- [126] X. Chen and S. Lathuilière, “Face aging via diffusion-based editing,” in *BMVC*, 2023.
- [127] F. Yang, S. Yang, M. A. Butt, J. van de Weijer *et al.*, “Dynamic prompt learning: Addressing cross-attention leakage for text-based image editing,” in *NeurIPS*, 2024.
- [128] B. Liu, C. Wang, T. Cao, K. Jia, and J. Huang, “Towards understanding cross and self-attention in stable diffusion for text-guided image editing,” in *CVPR*, 2024.
- [129] Q. Guo and T. Lin, “Focus on your instruction: Fine-grained and multi-instruction image editing by attention modulation,” *arXiv:2312.10113*, 2023.
- [130] N. Liu, S. Li, Y. Du, A. Torralba, and J. B. Tenenbaum, “Compositional visual generation with composable diffusion models,” in *ECCV*, 2022.
- [131] J. Ho and T. Salimans, “Classifier-free diffusion guidance,” *arXiv:2207.12598*, 2022.
- [132] S. Sadat, J. Buhmann, D. Bradley, O. Hilliges, and R. M. Weber, “CADS: unleashing the diversity of diffusion models through condition-annealed sampling,” in *ICLR*, 2024.
- [133] T. Kynkänniemi, M. Aittala, T. Karras, S. Laine, T. Aila, and J. Lehtinen, “Applying guidance in a limited interval improves sample and distribution quality in diffusion models,” *arXiv:2404.07724*, 2024.
- [134] J. Zhao, H. Zheng, C. Wang, L. Lan, and W. Yang, “Magicfu-

- sion: Boosting text-to-image generation performance by fusing diffusion models,” in *ICCV*, 2023.
- [135] T. Shirakawa and S. Uchida, “Noisecollage: A layout-aware text-to-image diffusion model based on noise cropping and merging,” in *CVPR*, 2024.
- [136] O. Bar-Tal, L. Yariv, Y. Lipman, and T. Dekel, “Multidiffusion: Fusing diffusion paths for controlled image generation,” in *ICML*, 2023.
- [137] Z. Luo, F. K. Gustafsson, Z. Zhao, J. Sjölund, and T. B. Schön, “Image restoration with mean-reverting stochastic differential equations,” in *ICML*, 2023.
- [138] B. Kawar, G. Väksman, and M. Elad, “Snips: Solving noisy inverse problems stochastically,” in *NeurIPS*, 2021.
- [139] B. Kawar, M. Elad, S. Ermon, and J. Song, “Denoising diffusion restoration models,” in *NeurIPS*, 2022.
- [140] S. Welker, H. N. Chapman, and T. Gerkmann, “Driftrec: Adapting diffusion models to blind jpeg restoration,” *IEEE TIP*, 2024.
- [141] Z. Yue, J. Wang, and C. C. Loy, “Resshift: Efficient diffusion model for image super-resolution by residual shifting,” in *NeurIPS*, 2024.
- [142] Y. Wang, W. Yang, X. Chen, Y. Wang, L. Guo, L.-P. Chau, Z. Liu, Y. Qiao, A. C. Kot, and B. Wen, “Sinsr: diffusion-based image super-resolution in a single step,” in *CVPR*, 2024.
- [143] M. Delbracio and P. Milanfar, “Inversion by direct iteration: An alternative to denoising diffusion for image restoration,” *TMLR*, 2023.
- [144] Y. Wang, J. Yu, and J. Zhang, “Zero-shot image restoration using denoising diffusion null-space model,” in *ICLR*, 2024.
- [145] P. Dhariwal and A. Nichol, “Diffusion models beat gans on image synthesis,” in *NeurIPS*, 2021.
- [146] H. Chung, B. Sim, D. Ryu, and J. C. Ye, “Improving diffusion models for inverse problems using manifold constraints,” in *NeurIPS*, 2022.
- [147] H. Chung, J. Kim, M. T. McCann, M. L. Klasky, and J. C. Ye, “Diffusion posterior sampling for general noisy inverse problems,” in *ICLR*, 2023.
- [148] O. Avrahami, D. Lischinski, and O. Fried, “Blended diffusion for text-driven editing of natural images,” in *CVPR*, 2022.
- [149] A. Voynov, K. Aberman, and D. Cohen-Or, “Sketch-guided text-to-image diffusion models,” in *ACM SIGGRAPH*, 2023.
- [150] G. Parmar, K. Kumar Singh, R. Zhang, Y. Li, J. Lu, and J.-Y. Zhu, “Zero-shot image-to-image translation,” in *ACM SIGGRAPH*, 2023.
- [151] J. Yu, Y. Wang, C. Zhao, B. Ghanem, and J. Zhang, “Freedom: Training-free energy-guided conditional diffusion model,” in *ICCV*, 2023.
- [152] A. Bansal, H.-M. Chu, A. Schwarzschild, S. Sengupta, M. Goldblum, J. Geiping, and T. Goldstein, “Universal guidance for diffusion models,” in *CVPR*, 2023.
- [153] J. Song, A. Vahdat, M. Mardani, and J. Kautz, “Pseudoinverse-guided diffusion models for inverse problems,” in *ICLR*, 2023.
- [154] L. Rout, N. Raoof, G. Daras, C. Caramanis, A. Dimakis, and S. Shakottai, “Solving linear inverse problems provably via posterior sampling with latent diffusion models,” in *NeurIPS*, 2024.
- [155] H. Chung, J. Kim, S. Kim, and J. C. Ye, “Parallel diffusion models of operator and image for blind inverse problems,” in *CVPR*, 2023.
- [156] B. Fei, Z. Lyu, L. Pan, J. Zhang, W. Yang, T. Luo, B. Zhang, and B. Dai, “Generative diffusion prior for unified image restoration and enhancement,” in *CVPR*, 2023.
- [157] X. Liu, D. H. Park, S. Azadi, G. Zhang, A. Chopikyan, Y. Hu, H. Shi, A. Rohrbach, and T. Darrell, “More control for free! image synthesis with semantic diffusion guidance,” in *WACV*, 2023.
- [158] G. Kwon and J. C. Ye, “Diffusion-based image translation using disentangled style and content representation,” in *ICLR*, 2023.
- [159] J. Singh, S. Gould, and L. Zheng, “High-fidelity guided image synthesis with latent diffusion models,” in *CVPR*, 2023.
- [160] G. Luo, T. Darrell, O. Wang, D. B. Goldman, and A. Holynski, “Readout guidance: Learning control from diffusion features,” in *CVPR*, 2024.
- [161] S. Mo, F. Mu, K. H. Lin, Y. Liu, B. Guan, Y. Li, and B. Zhou, “Freecontrol: Training-free spatial control of any text-to-image diffusion model with any condition,” in *CVPR*, 2024.
- [162] Y. Lin, S. Zhang, X. Yang, X. Wang, and Y. Shi, “Regeneration learning of diffusion models with rich prompts for zero-shot image translation,” *arXiv:2305.04651*, 2023.
- [163] G. Y. Park, J. Kim, B. Kim, S. W. Lee, and J. C. Ye, “Energy-based cross attention for bayesian context update in text-to-image diffusion models,” in *NeurIPS*, 2024.
- [164] M. Chen, I. Laina, and A. Vedaldi, “Training-free layout control with cross-attention guidance,” in *WACV*, 2024.
- [165] D. Epstein, A. Jabri, B. Poole, A. Efros, and A. Holynski, “Diffusion self-guidance for controllable image generation,” in *NeurIPS*, 2024.
- [166] C. Mou, X. Wang, J. Song, Y. Shan, and J. Zhang, “Diffeditor: Boosting accuracy and flexibility on diffusion-based image editing,” in *CVPR*, 2024.
- [167] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, “Score-based generative modeling through stochastic differential equations,” in *ICLR*, 2021.
- [168] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool, “Repaint: Inpainting using denoising diffusion probabilistic models,” in *CVPR*, 2022.
- [169] J. Choi, S. Kim, Y. Jeong, Y. Gwon, and S. Yoon, “ILVR: conditioning method for denoising diffusion probabilistic models,” in *ICCV*, 2021.
- [170] G. Couairon, J. Verbeek, H. Schwenk, and M. Cord, “Diffedit: Diffusion-based semantic image editing with mask guidance,” in *ICLR*, 2023.
- [171] H. Chung, B. Sim, and J. C. Ye, “Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction,” in *CVPR*, 2022.
- [172] O. Patashnik, D. Garabi, I. Azuri, H. Averbuch-Elor, and D. Cohen-Or, “Localizing object-level shape variations with text-to-image diffusion models,” in *ICCV*, 2023.
- [173] Q. Wang, B. Zhang, M. Birsak, and P. Wonka, “Instructedit: Improving automatic masks for diffusion-based image editing with user instructions,” *arXiv:2305.18047*, 2023.
- [174] Y. Lin, Y.-W. Chen, Y.-H. Tsai, L. Jiang, and M.-H. Yang, “Text-driven image editing via learnable regions,” in *CVPR*, 2024.
- [175] N. Huang, F. Tang, W. Dong, T.-Y. Lee, and C. Xu, “Region-aware diffusion for zero-shot text-driven image editing,” *arXiv:2302.11797*, 2023.
- [176] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, “Deep unsupervised learning using nonequilibrium thermodynamics,” in *ICML*, 2015.
- [177] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” in *NeurIPS*, 2020.
- [178] T. Karras, M. Aittala, T. Aila, and S. Laine, “Elucidating the design space of diffusion-based generative models,” in *NeurIPS*, 2022.
- [179] D. Chen, Z. Zhou, C. Wang, C. Shen, and S. Lyu, “On the trajectory regularity of ode-based diffusion sampling,” in *ICML*, 2024.
- [180] C. Zhang, C. Zhang, M. Zhang, and I. S. Kweon, “Text-to-image diffusion models in generative ai: A survey,” *arXiv:2303.07909*, 2023.
- [181] Y. Li, K. Zhou, W. X. Zhao, and J.-R. Wen, “Diffusion models for non-autoregressive text generation: a survey,” in *IJCAI*, 2023.
- [182] Y. Huang, J. Huang, Y. Liu, M. Yan, J. Lv, J. Liu, W. Xiong, H. Zhang, S. Chen, and L. Cao, “Diffusion model-based image editing: A survey,” *arXiv:2402.17525*, 2024.

- [183] F.-A. Croitoru, V. Hondru, R. T. Ionescu, and M. Shah, “Diffusion models in vision: A survey,” *IEEE TPAMI*, 2023.
- [184] R. Po, W. Yifan, V. Golyanik, K. Aberman, J. T. Barron, A. H. Bermano, E. R. Chan, T. Dekel, A. Holynski, A. Kanazawa *et al.*, “State of the art on diffusion models for visual computing,” *arXiv:2310.07204*, 2023.
- [185] X. Shuai, H. Ding, X. Ma, R. Tu, Y.-G. Jiang, and D. Tao, “A survey of multimodal-guided image editing with text-to-image diffusion models,” *arXiv:2406.14555*, 2024.
- [186] H. Cao, C. Tan, Z. Gao, Y. Xu, G. Chen, P.-A. Heng, and S. Z. Li, “A survey on generative diffusion models,” *IEEE Transactions on Knowledge and Data Engineering*, 2024.
- [187] J. Song, C. Meng, and S. Ermon, “Denoising diffusion implicit models,” in *ICLR*, 2021.
- [188] B. Oksendal, *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013.
- [189] D. Chen, Z. Zhou, J.-P. Mei, C. Shen, C. Chen, and C. Wang, “A geometric perspective on diffusion models,” *arXiv:2305.19947*, 2023.
- [190] Q. Zhang and Y. Chen, “Fast sampling of diffusion models with exponential integrator,” in *ICLR*, 2023.
- [191] P. Vincent, “A connection between score matching and denoising autoencoders,” *Neural Computation*, 2011.
- [192] S. Lyu, “Interpretation and generalization of score matching,” in *UAI*, 2009.
- [193] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *MICCAI*, 2015.
- [194] Y. Song and S. Ermon, “Generative modeling by estimating gradients of the data distribution,” in *NeurIPS*, 2019.
- [195] ———, “Improved techniques for training score-based generative models,” in *NeurIPS*, 2020.
- [196] A. Q. Nichol and P. Dhariwal, “Improved denoising diffusion probabilistic models,” in *ICML*, 2021.
- [197] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Mininderer, G. Heigold, S. Gelly *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv:2010.11929*, 2020.
- [198] R. Li, W. Li, Y. Yang, H. Wei, J. Jiang, and Q. Bai, “Swinv2-imagen: Hierarchical vision transformer diffusion models for text-to-image generation,” *NCA*, 2023.
- [199] X. Yang, S.-M. Shih, Y. Fu, X. Zhao, and S. Ji, “Your vit is secretly a hybrid discriminative-generative diffusion model,” *arXiv:2208.07791*, 2022.
- [200] F. Bao, S. Nie, K. Xue, Y. Cao, C. Li, H. Su, and J. Zhu, “All are worth words: A vit backbone for diffusion models,” in *CVPR*, 2023.
- [201] S. Sheynin, O. Ashual, A. Polyak, U. Singer, O. Gafni, E. Nachmani, and Y. Taigman, “knn-diffusion: Image generation via large-scale retrieval,” in *ICLR*, 2023.
- [202] Z. Tang, S. Gu, J. Bao, D. Chen, and F. Wen, “Improved vector quantized diffusion models,” *arXiv:2205.16007*, 2022.
- [203] S. Chai, L. Zhuang, and F. Yan, “Layoutdm: Transformer-based diffusion model for layout generation,” in *CVPR*, 2023.
- [204] S. Pan, T. Wang, R. L. Qiu, M. Axente, C.-W. Chang, J. Peng, A. B. Patel, J. Shelton, S. A. Patel, J. Roper *et al.*, “2d medical image synthesis using transformer-based denoising diffusion probabilistic model,” *Physics in Medicine & Biology*, 2023.
- [205] D. Zheng, X.-M. Wu, S. Yang, J. Zhang, J.-F. Hu, and W.-S. Zheng, “Selective hourglass mapping for universal image restoration based on diffusion model,” in *CVPR*, 2024.
- [206] Q. Zheng, L. Zheng, Y. Guo, Y. Li, S. Xu, J. Deng, and H. Xu, “Self-adaptive reality-guided diffusion for artifact-free super-resolution,” in *CVPR*, 2024.
- [207] S. Li, C. Chen, and H. Lu, “Moecontroller: Instruction-based arbitrary image manipulation with mixture-of-expert controllers,” *arXiv:2309.04372*, 2023.
- [208] C. Qin, S. Zhang, N. Yu, Y. Feng, X. Yang, Y. Zhou, H. Wang, J. C. Niebles, C. Xiong, S. Savarese *et al.*, “Unicontrol: a unified diffusion model for controllable visual generation in the wild,” in *NeurIPS*, 2023.
- [209] S. Yang, X. Chen, and J. Liao, “Uni-paint: A unified framework for multimodal image inpainting with pretrained diffusion model,” in *ACM MM*, 2023.
- [210] S. Zhao, D. Chen, Y.-C. Chen, J. Bao, S. Hao, L. Yuan, and K.-Y. K. Wong, “Uni-controlnet: All-in-one control to text-to-image diffusion models,” in *NeurIPS*, 2024.
- [211] L. Ran, X. Cun, J.-W. Liu, R. Zhao, S. Zijie, X. Wang, J. Keppo, and M. Z. Shou, “X-adapter: Adding universal compatibility of plugins for upgraded diffusion model,” in *CVPR*, 2024.
- [212] Z. Jiang, C. Mao, Y. Pan, Z. Han, and J. Zhang, “Scedit: Efficient and controllable image diffusion generation via skip connection editing,” *arXiv:2312.11392*, 2023.
- [213] J. Zeng, D. Song, W. Nie, H. Tian, T. Wang, and A.-A. Liu, “Cat-dm: Controllable accelerated virtual try-on with diffusion model,” in *CVPR*, 2024.
- [214] J. Kim, G. Gu, M. Park, S. Park, and J. Choo, “Stableviton: Learning semantic correspondence with latent diffusion model for virtual try-on,” in *CVPR*, 2024.
- [215] J. Tian, L. Aggarwal, A. Colaco, Z. Kira, and M. Gonzalez-Franco, “Diffuse attend and segment: Unsupervised zero-shot segmentation using stable diffusion,” in *CVPR*, 2024.
- [216] Z. Yang, G. Ding, W. Wang, H. Chen, B. Zhuang, and C. Shen, “Object-aware inversion and reassembly for image editing,” in *ICLR*, 2023.
- [217] H. Lee, M. Kang, and B. Han, “Conditional score guidance for text-driven image-to-image translation,” in *NeurIPS*, 2024.
- [218] A. Van Den Oord, O. Vinyals *et al.*, “Neural discrete representation learning,” in *NeurIPS*, 2017.
- [219] J. D. M.-W. C. Kenton and L. K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *NAACL-HLT*, 2019.
- [220] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, “Learning transferable visual models from natural language supervision,” in *ICML*, 2021.
- [221] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *Journal of machine learning research*, 2020.
- [222] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin transformer: Hierarchical vision transformer using shifted windows,” in *ICCV*, 2021.
- [223] J. Li, D. Li, S. Savarese, and S. Hoi, “Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models,” in *ICML*, 2023.
- [224] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, “Arcface: Additive angular margin loss for deep face recognition,” in *CVPR*, 2019.
- [225] H. Liu, C. Li, Q. Wu, and Y. J. Lee, “Visual instruction tuning,” in *NeurIPS*, 2024.
- [226] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, “Language models are few-shot learners,” *NeurIPS*, 2020.
- [227] J. Li, D. Li, C. Xiong, and S. Hoi, “Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation,” in *ICML*, 2022.
- [228] D. F. Shanno, “Conditioning of quasi-newton methods for function minimization,” *Mathematics of computation*, 1970.
- [229] S. Kim, W. Jang, H. Kim, J. Kim, Y. Choi, S. Kim, and G. Lee, “User-friendly image editing with minimal text input: Leveraging captioning and injection techniques,” *arXiv:2306.02717*, 2023.
- [230] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, “Segment anything,” in *ICCV*, 2023.

- [231] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, “Lora: Low-rank adaptation of large language models,” *arXiv:2106.09685*, 2021.
- [232] L. Han, S. Wen, Q. Chen, Z. Zhang, K. Song, M. Ren, R. Gao, A. Stathopoulos, X. He, Y. Chen *et al.*, “Proxedit: Improving tuning-free real image editing with proximal guidance,” in *WACV*, 2024.
- [233] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu, “Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps,” in *NeurIPS*, 2022.
- [234] ———, “Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models,” *arXiv:2211.01095*, 2022.
- [235] Y. Li, H. Wang, Q. Jin, J. Hu, P. Chemerys, Y. Fu, Y. Wang, S. Tulyakov, and J. Ren, “Snapfusion: Text-to-image diffusion model on mobile devices within two seconds,” in *NeurIPS*, 2024.
- [236] Y. Zhao, Y. Xu, Z. Xiao, and T. Hou, “Mobilediffusion: Subsecond text-to-image generation on mobile devices,” *arXiv:2311.16567*, 2023.
- [237] Z. Zhou, D. Chen, C. Wang, and C. Chen, “Fast ode-based sampling for diffusion models in around 5 steps,” in *CVPR*, 2024.
- [238] T. Salimans and J. Ho, “Progressive distillation for fast sampling of diffusion models,” in *ICLR*, 2022.
- [239] C. Meng, R. Rombach, R. Gao, D. Kingma, S. Ermon, J. Ho, and T. Salimans, “On distillation of guided diffusion models,” in *CVPR*, 2023.
- [240] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever, “Consistency models,” in *ICML*, 2023.
- [241] D. Chen, J. Mei, Y. Zhang, C. Wang, Z. Wang, Y. Feng, and C. Chen, “Cross-layer distillation with semantic calibration,” in *AAAI*, 2021.
- [242] D. Chen, J.-P. Mei, H. Zhang, C. Wang, Y. Feng, and C. Chen, “Knowledge distillation with the reused teacher classifier,” in *CVPR*, 2022.
- [243] B.-K. Kim, H.-K. Song, T. Castells, and S. Choi, “Bk-sdm: A lightweight, fast, and cheap version of stable diffusion,” *arXiv:2305.15798*, 2023.
- [244] C. Schuhmann, R. Vencu, R. Beaumont, R. Kaczmarczyk, C. Mullis, A. Katta, T. Coombes, J. Jitsev, and A. Komatsuzaki, “Laion-400m: Open dataset of clip-filtered 400 million image-text pairs,” *arXiv:2111.02114*, 2021.
- [245] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman *et al.*, “Laion-5b: An open large-scale dataset for training next generation image-text models,” in *NeurIPS*, 2022.
- [246] E. Agustsson and R. Timofte, “Ntire 2017 challenge on single image super-resolution: Dataset and study,” in *CVPR*, 2017.
- [247] S. Nah, T. Hyun Kim, and K. Mu Lee, “Deep multi-scale convolutional neural network for dynamic scene deblurring,” in *CVPR*, 2017.
- [248] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in *CVPR*, 2019.
- [249] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *ECCV*, 2014.
- [250] H. Caesar, J. Uijlings, and V. Ferrari, “Coco-stuff: Thing and stuff classes in context,” in *CVPR*, 2018.
- [251] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, “Scene parsing through ade20k dataset,” in *CVPR*, 2017.
- [252] Z. Yuan, L. Li, Z. Wang, and X. Zhang, “Watermarking for stable diffusion models,” *IEEE Internet of Things Journal*, 2024.
- [253] Y. Cui, J. Ren, H. Xu, P. He, H. Liu, L. Sun, Y. Xing, and J. Tang, “Diffusionshield: A watermark for copyright protection against generative diffusion models,” *arXiv:2306.04642*, 2023.
- [254] Y. Wen, J. Kirchenbauer, J. Geiping, and T. Goldstein, “Tree-ring watermarks: Fingerprints for diffusion images that are invisible and robust,” *arXiv:2305.20030*, 2023.
- [255] X. Shen, C. Du, T. Pang, M. Lin, Y. Wong, and M. Kankanhalli, “Finetuning text-to-image diffusion models for fairness,” in *ICLR*, 2023.
- [256] S. Um, S. Lee, and J. C. Ye, “Don’t play favorites: Minority guidance for diffusion models,” in *ICLR*, 2024.
- [257] H. Li, C. Shen, P. Torr, V. Tresp, and J. Gu, “Self-discovering interpretable diffusion latent directions for responsible text-to-image generation,” in *CVPR*, 2024.
- [258] P. Schramowski, M. Brack, B. Deisereth, and K. Kersting, “Safe latent diffusion: Mitigating inappropriate degeneration in diffusion models,” in *CVPR*, 2023.