Bayesian Power Steering: An Effective Approach for Domain Adaptation of Diffusion Models

Ding Huang [1] **Ting Li** [1] **Jian Huang** [1]

Abstract

We propose a Bayesian framework for fine-tuning large diffusion models with a novel network structure called Bayesian Power Steering (BPS) [2]. We clarify the meaning behind adaptation from a *large probability space* to a *small probability space* and explore the task of fine-tuning pre-trained models using learnable modules from a Bayesian perspective. BPS extracts task-specific knowledge from a pre-trained model's learned prior distribution. It efficiently leverages large diffusion models, differentially intervening different hidden features with a head-heavy and foot-light configuration. Experiments highlight the superiority of BPS over contemporary methods across a range of tasks even with limited amount of data. Notably, BPS attains an FID score of 10.49 under the sketch condition on the COCO17 dataset.

1. Introduction

The advent of diffusion models (Ho et al., 2020; Song et al., 2020b) and their extensions (Song et al., 2020a; Nichol & Dhariwal, 2021; Huang et al., 2023), has enabled effective learning of intricate probability measures for diverse data types, including images (Ho et al., 2022; Rombach et al., 2022; Saharia et al., 2022; Ho et al., 2022), audio (Kong et al., 2020), and 3D bioimaging data (Luo & Hu, 2021; Poole et al., 2022; Shi et al., 2023; Pinaya et al., 2022). For these generative models, the quantity of training data plays a crucial role in influencing both the precision of probability measure estimation and the generalization capacity, enabling them to extrapolate effectively within the probability space. Particularly, in computer vision, significant efforts have been devoted to developing large-scale diffusion models. For instance, the Stable Diffusion (SD, Rombach et al. (2022)) is trained utilizing the LAION-5B dataset (Schuhmann et al., 2022), a large publicly available text-image dataset, with a staggering magnitude of 585 billion.

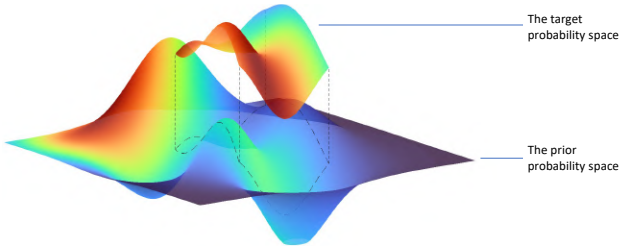


Figure 1. The relationship between the probability spaces learned by the pre-trained model (the prior probability space or *large probability space*) and the target task (the target probability space or *small probability space*). Different colors represent different densities at various locations on the manifold.

The diversity and abundance of data contributes to the exceptional generative capabilities of large-scale models. They can capture intricate details in the large probability space of images. However, numerous data-driven modeling tasks typically focus on a specific subset of the entire image space, as illustrated in Figure 1. Moreover, the size of the training data available for these tasks is considerably smaller compared to expansive datasets such as LAION-5B. This limitation is especially evident in user-customized scenarios, where the available number of samples provided by users is often limited. Consequently, learning this *smaller probability space* poses a significant challenge.

Pre-trained large models, which encapsulate information across the *large probability space* and exhibit exceptional generalization abilities, offer a potential solution to the challenge posed by limited training data sizes. This prompts us to ask the question: can the judicious deployment of such pre-trained large-scale models, in conjunction with carefully curated specialized datasets, effectively facilitate the transition from a *large probability space* to a *small probability space*?

[1]Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong, China. Correspondence to: Ting Li <tingeric.li@polyu.edu.hk>, Jian Huang <j.huang@polyu.edu.hk>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

[2]Code and models are available at https://github.com/DingDing33/BPS-v1-1.

A related topic is **domain adaptation for generative tasks**. Originating from the field of transfer learning, this concept has gradually extended its application to generation tasks, particularly aimed at transferring a broader knowledge base acquired from publicly available data to the task-specific distribution (Li et al., 2021; Harder et al., 2022; Lyu et al., 2023; Kurakin et al., 2023). Especially in privacy protection and medical data domains, with a specific focus on creating infinite amounts of synthetic data to maximise the downstream prediction performance (Sagers et al., 2023; Ghalebikesabi et al., 2023; Li et al., 2023; Tang et al., 2024). Another notable application lies in customizing scenarios for users. Textual inversion (Gal et al., 2022), DreamBooth (Ruiz et al., 2023), LayoutDiffuse (Cheng et al., 2023) and StyleDrop (Sohn et al., 2023) are exemplary techniques to capture the specific domain (semantic concepts, patterns, styles, etc.), leveraging a small set of user-provided example images.

Other efforts have been devoted to **conditional control for text-to-image diffusion models** for achieving customized image generation. Text-based control methods, as explored by Brooks et al. (2023); Hertz et al. (2022); Gafni et al. (2022); Kawar et al. (2023); Parmar et al. (2023); Chefer et al. (2023); Orgad et al. (2023); and Couairon et al. (2022), are centered around the adjustment of prompts, manipulation of CLIP features, and modification of cross-attention mechanisms. The methods of (Ho & Salimans, 2022; Hong et al., 2023; Nichol et al., 2021; Brack et al., 2023) combine diverse prompts to achieve desired outcomes. For control conditions other than text, Dhariwal & Nichol (2021) introduce a time-dependent classifier, which requires a separate training procedure and leads to a performance decline (Ghalebikesabi et al., 2023) due to a decreasing signal-to-noise ratio over time. Concurrent endeavors including T2I-Adapter (Mou et al., 2023) and ControlNet (Zhang et al., 2023) employ fine-tuning techniques on large-scale diffusion models, which enable precise spatial control over image generation and establish the state of the art (SOTA).

In this paper, we first mathematically formulate the problem, proposing a framework for transition from a *large probability space* to a *small probability space* through its transformation into a conditional generative problem. To tackle this problem, we propose a Bayesian fine-tuning framework along with a novel network structure called Bayesian Power Steering (BPS), designed to shift the pretrained diffusion system towards the data support specified by the given conditions.

Our extensive experiments demonstrate that BPS, with significantly reduced amounts of data, generates samples that align with the support of the dataset and achieves comparable generative quality to large-scale models. Our ablation studies confirm the robustness and scalability of BPS across datasets of varying sizes. Experiments validate the effectiveness of the proposed model, the necessity of its components, and , and provide a comparative analysis against strong conditional image generation baselines. Notably,the proposed method surpasses state-of-art methods, achieving the FID score of 10.49 with sketch condition on the COCO17 dataset (Lin et al., 2014).

2. Preliminary: Stable Diffusion

Throughout this paper, we adopt the Stable Diffusion (Rombach et al., 2022) as the pre-trained large-scale model. Initially, the image data $X_0 \in \mathbb{R}^{d_{\text{img}}}(d_{\text{img}} = 512^2 \times 3)$ is mapped to a lower-dimensional latent probability space $\mathcal{Z} := (\Omega, \mathfrak{F}, \mathcal{P}) \subset \mathbb{R}^d (d = 64^2 \times 4)$ via the pre-trained autoencoder (Esser et al., 2021), while related text prompts are encoded as $C_{\text{text}} \in \mathbb{R}^{k_{\text{text}}}$ by CLIP (Radford et al., 2021). Subsequently, the diffusion process (Ho et al., 2020) is employed to generate representations of the image in the latent space with condition C_{text}. Finally, the image is reconstructed using the pre-trained decoder.

The forward diffusion process in the latent space $\mathbb{R}^d$ is expressed as $\{Z_t\}_{t=0}^T$ with $t \in [T] := \{1, ..., T\}$:

$$Z_t = \sqrt{\bar{\alpha}_t} Z_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\eta},\ \boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d), \tag{1}$$

where $Z_0 \in \mathcal{Z}$ denotes the generative object, and $\{\bar{\alpha}_t\}_{t=1}^T$ is a strictly decreasing sequence within the interval $(0, 1)$. In the reverse process, conditional modeling is accomplished by manipulating the direction of the score function with conditions. The backward process $\{\tilde{Z}_t^{C_{\text{text}}}\}_{t=1}^T$ starts from $\tilde{Z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ with the following iteration:

$$\begin{aligned}\tilde{Z}_{t-1}^{C_{\text{text}}} =& \frac{1}{1-\beta_t}\left(\tilde{Z}_t^{C_{\text{text}}} - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}^*(\tilde{Z}_t^{C_{\text{text}}}, t, C_{\text{text}})\right) \\ &+ \sigma_t \boldsymbol{\eta},\end{aligned} \tag{2}$$

, here $\beta_t := 1 - \bar{\alpha}_t/\bar{\alpha}_{t-1}$, $\sigma_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$. According to Tweedie's formula (Efron, 2011), the denoise function $\boldsymbol{\epsilon}^*(\mathbf{z}_t, t, \mathbf{c}_{\text{text}})$ can be expressed as

$$\begin{aligned}\boldsymbol{\epsilon}^*(\mathbf{z}_t, t, \mathbf{c}_{\text{text}}) :&= -\sqrt{1-\bar{\alpha}_t}\nabla \log p(Z_t = \mathbf{z}_t \mid C_{\text{text}} = \mathbf{c}_{\text{text}}) \\ &= \mathbb{E}[\boldsymbol{\eta} \mid Z_t = \mathbf{z}_t, C_{\text{text}} = \mathbf{c}_{\text{text}}].\end{aligned} \tag{3}$$

The denoise function is parameterized through a neural network, denoted as $\boldsymbol{\epsilon}_{\hat{\boldsymbol{\theta}}}$. As demonstrated in equation (2), precise control over the sampling process can be attained by conducting fine-tuning on the pre-trained denoise function $\boldsymbol{\epsilon}_{\hat{\boldsymbol{\theta}}}$.

3. General Setup

Our objective is to fine-tune the pre-trained denoise function $\boldsymbol{\epsilon}_{\hat{\boldsymbol{\theta}}}$ with learnable modules, aiming to extract a *small probability space* from a *large probability space*.

3.1. Problem Formulation

We first establish the precise mathematical definitions for both *large* and *small latent probability spaces*. Let Z_0 defined in the latent probability space $\mathcal{Z} := (\Omega, \mathfrak{F}, \mathcal{P})$ be the generative target of the pre-trained model. Our focus lies on a *small latent probability space* residing within a non-zero measurable set $\Delta \in \mathfrak{F}$ (Fig.1). This space corresponds to the trace of $\mathcal{Z}$ on Δ and is denoted as $\mathcal{Z}_\Delta := (\Delta, \Delta \cap \mathfrak{F}, \mathcal{P}_\Delta)$. Specifically, the target probability measure is defined by $\mathcal{P}_\Delta(E) := \mathcal{P}(E)/\mathcal{P}(\Delta)$, for all $E \in \Delta \cap \mathfrak{F}$.

According to the following lemma, the target domain Δ can be described using a suitable condition $\mathbf{c} \in \mathbb{R}^k$.

Lemma 3.1. *(Chung, 2001) If $\Delta \in \mathfrak{F}$, then there exists some measurable function $\boldsymbol{\psi}(\cdot)$ such that $\Delta(\omega) = \boldsymbol{\psi}(Z_0)$ for any $\Delta(\omega) := \omega \in \Delta \cap \mathfrak{F}$.*

By introducing a function $\boldsymbol{\psi} : \Omega \to \mathbb{R}^k$ to describe the relationship $\mathbf{c} = \boldsymbol{\psi}(\mathbf{z}_0)$, Δ can be defined as the union of sets $\Delta = \cup_{\mathbf{c} \in \boldsymbol{\psi}(\Delta)} \{\mathbf{z}_0 : \boldsymbol{\psi}(\mathbf{z}_0) = \mathbf{c}\}$. Consequently, when the data $\mathbf{z}_0 \in \Delta$ aligns with suitable conditions $\mathbf{c}$, the task of learning a "small distribution" is formulated as learning a probability measure of $Z_0 \mid C$, where C is a random variable defined in $\mathcal{C} := (\boldsymbol{\psi}(\Delta), \boldsymbol{\psi}(\mathfrak{F}), \mathcal{P}') \subseteq \mathbb{R}^k$.

The form of conditions can vary widely. For example, in Text-to-Image Diffusion Models, textual descriptions, denoted as C_{text}, are employed to guide the generative process. However, given that a single image can correspond to multiple descriptions, establishing a direct mapping ψ from image to text alone is not practical. To address this, additional descriptors C_{add}, such as edge, depth, or attention information, can be incorporated based on the specific goals of the model. These supplementary controls, which can be numerous, aid in pinpointing the desired target support Δ. Consequently, the condition in this context is represented as a k-dimensional vector $C := (C_{\text{text}}^\top, C_{\text{add}}^\top)^\top$, integrating both textual and additional descriptive elements.

The proposed BPS method, which is formally introduced in Section 4, is based on the above formulation for fine-tuning the pre-trained denoise function $\boldsymbol{\epsilon}_{\hat{\boldsymbol{\theta}}}$ using independent paired samples from $\mathcal{X} \times \mathcal{C}$.

3.2. Bayesian Formulation

Suppose condition $(\mathbf{c}_{\text{text}}, \mathbf{c}_{\text{add}})$ provides a detailed characterization of the target domain, then our primary objective is to learn the integrated denoise function, denoted as

$$\bar{\boldsymbol{\epsilon}}^*(\mathbf{z}_t, t, \mathbf{c}_{\text{text}}, \mathbf{c}_{\text{add}}) := \mathbb{E}[\boldsymbol{\eta} \mid \mathbf{z}_t, \mathbf{c}_{\text{text}}, \mathbf{c}_{\text{add}}].$$

To take advantage of the pretrained model, we bridge $\bar{\boldsymbol{\epsilon}}^*(\mathbf{z}_t, t, \mathbf{c}_{\text{text}}, \mathbf{c}_{\text{add}})$ and $\boldsymbol{\epsilon}^*(\mathbf{z}_t, t, \mathbf{c}_{\text{text}})$, using Tweedie's formula (Efron, 2011) and Bayes' theorem as follows. First, Tweedie's formula gives

$$\bar{\boldsymbol{\epsilon}}^*(\mathbf{z}_t, t, \mathbf{c}_{\text{text}}, \mathbf{c}_{\text{add}}) = -\sqrt{1-\bar{\alpha}_t}\, \nabla \log p(\mathbf{z}_t \mid \mathbf{c}_{\text{text}}, \mathbf{c}_{\text{add}}).$$

By Bayes' theorem, we have

$$p(\mathbf{z}_t \mid \mathbf{c}_{\text{text}}, \mathbf{c}_{\text{add}}) = \frac{p(\mathbf{c}_{\text{add}} \mid \mathbf{z}_t, \mathbf{c}_{\text{text}})}{p(\mathbf{c}_{\text{add}} \mid \mathbf{c}_{\text{text}})} p(\mathbf{z}_t \mid \mathbf{c}_{\text{text}}). \quad (4)$$

Taking the logarithm across (4), differentiating with respect to $\mathbf{z}_t$, and then using the definitions of $\bar{\boldsymbol{\epsilon}}^*(\mathbf{z}_t, t, \mathbf{c}_{\text{text}}, \mathbf{c}_{\text{add}})$ given above and $\boldsymbol{\epsilon}^*(\mathbf{z}_t, t, \mathbf{c}_{\text{text}})$ from Equation (3), we have

$$\begin{aligned} &\bar{\boldsymbol{\epsilon}}^*(\mathbf{z}_t, t, \mathbf{c}_{\text{text}}, \mathbf{c}_{\text{add}}) \quad (5)\\ &= -\sqrt{1-\bar{\alpha}_t}[\nabla \log p(\mathbf{c}_{\text{add}} \mid \mathbf{z}_t, \mathbf{c}_{\text{text}}) + \nabla \log p(\mathbf{z}_t \mid \mathbf{c}_{\text{text}})] \\ &= -\sqrt{1-\bar{\alpha}_t} \nabla \log p(\mathbf{c}_{\text{add}} \mid \mathbf{z}_t, \mathbf{c}_{\text{text}}) + \boldsymbol{\epsilon}^*(\mathbf{z}_t, t, \mathbf{c}_{\text{text}}). \end{aligned}$$

Detailed derivations are deferred to Appendix A. Below, we denote

$$M(\mathbf{z}_t, t, \mathbf{c}_{\text{text}}, \mathbf{c}_{\text{add}}) = -\sqrt{1-\bar{\alpha}_t} \nabla \log p(\mathbf{c}_{\text{add}} \mid \mathbf{z}_t, \mathbf{c}_{\text{text}}). \quad (6)$$

The integrated function $\bar{\boldsymbol{\epsilon}}^*$ can be interpreted as the *posterior denoise function* corresponding to the *prior denoise function* $\boldsymbol{\epsilon}^*$ for the pretrained model. It is obtained by combining the pretrained denoise function $\boldsymbol{\epsilon}^*$ with a time-dependent "steering gear" M.

We emphasizes the role of the learnable modules as time-dependent "steering gears" $\{M(\mathbf{z}_t, t, \mathbf{c}_{\text{text}}, \mathbf{c}_{\text{add}}), t = 1, \ldots, T\}$ within the architecture, guiding the system towards directions with a high probability density of satisfying condition C. However, such externally appended fine-tuning structure introduces the problem of high-dimensional input-output, thereby increasing the computational burden during training. To address it, we integrate the information of $\mathbf{c}_{\text{add}}$ into the pretrained denoising function by realizing this corollary in the feature space, discussed in the following subsection.

3.3. Integration Strategy

Our study utilizes a pretrained model $\boldsymbol{\epsilon}_{\hat{\boldsymbol{\theta}}}$ based on the U-net architecture (Ronneberger et al., 2015). This architecture consists of an encoder (E.), a middle block (MB.), a skip-connected decoder (D.), and skip-connections between the encoder and decoder (E-D.). These components sequentially extract information from the input, yielding distinctive levels of feature space.

Based on the insights from equation 5, we explore potential schemes for integrating residual structures across various hierarchical levels of the feature space, as outlined in Table 1. Notably, mode ME-D is adopted in ControlNet (Zhang et al.,

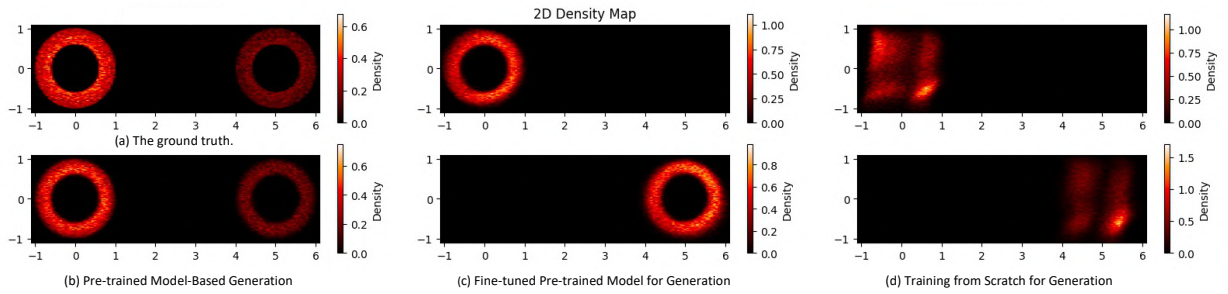


Figure 2. The simulation of 2D measure. The pre-trained model is trained with 500K samples, and the generative results are shown in Fig. (b). In Figures (c) and (d), labeled subsets of 100 samples were utilized in the experiments. Fig. (c) illustrates the generation outcomes achieved through fine-tuning the pre-trained model with 2.5K epochs, whereas Fig. (d) displays the results obtained by training the same model from scratch with 20K epochs. The above legends represent the sampling of 5K samples on either the simulated or real measure.

Table 1. Sites of modular integration with pre-trained models.

MODE	E.	MB.	D.	E-D.
ALL	√	√	√	√
EMD	√	√	√	×
E	√	×	×	×
EM	√	√	×	×
D	×	×	√	×
MD	×	√	√	×
E-D	×	×	×	√
ME-D	×	√	×	√
M	×	√	×	×

2023) and mode EM is adopted in T2I-Adapter Mou et al. (2023). We conduct a comparative analysis of the performance exhibited by different integration modes, considering the variables t and $\mathbf{c}_{\text{add}}$ as inputs, in following generation tasks.

Generation of multimodal 2D data. We use this example to demonstrate the formulation in subsection 3.1 and provide a simple illustration of the impact of conditional information on the denoising direction offset. As show in Fig. 2(a), the complete support comprises two rings, and the target domain Δ is a specified ring. We construct a U-net tailored for this vector data and pre-train a large-scale model. The generation results are presented in Fig. 2(b).

We quantify the effectiveness of integration modes by the accuracy of the support set to which the generated samples belong. According to Fig.3, the generative performance can be classified into three tiers. The first tier (ALL, EMD, and EM) achieves the highest accuracy and the fastest convergence rate. The second tier (E, EM, and ME-D) and the third tier (M, MD, and D) exhibit relatively worse performance. Furthermore, the comparison of mode E and EM (D and MD/E-D and ME-D) highlights the critical impact of injecting conditional information in the intermediate blocks. Moreover, the analysis of modes E, M, and D elucidates that introducing conditional information at different levels of the feature space leads to varying degrees of influence on the output gradient offset. Notably, injecting information at earlier stages exerts a more significant impact on the directional offset.

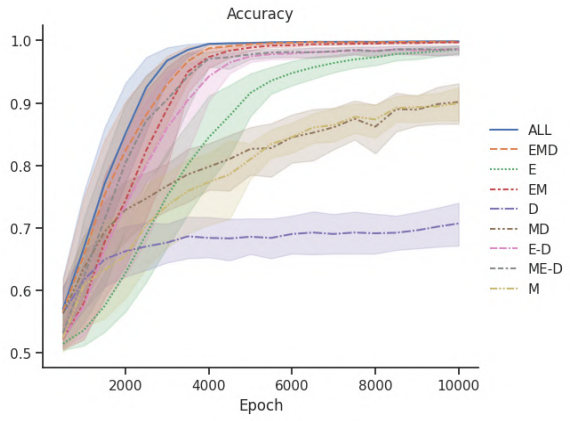


Figure 3. Classification accuracy of generated samples from distinct integration modes with $n = 12$ training samples. For each mode, five experiments are conducted.

The robustness of various integration models concerning sample size and the number of iterations is shown in Fig.4. The first tier modes (ALL, EMD, and EM) consistently demonstrate superior performance and convergence speed.

Segmentation-to-image. For this task, the performance is assessed using the ADE20K dataset (Zhou et al., 2017). The conditioning fidelity is evaluated through Mean Intersection-over-Union (mIoU), and the state-of-the-art segmentation method OneFormer (Jain et al., 2023) achieves the mIoU of 0.58.

We utilize the pre-trained SD as the backbone and exam mode ALL, EMD, EM, MC-E, and MD. We generate images using segmentations from the ADE20K validation set and then feed the generative results to OneFormer for segmentation detection and computation of reconstructed IoUs.

From Fig.4, these modes exhibit superior performance in different tiers.

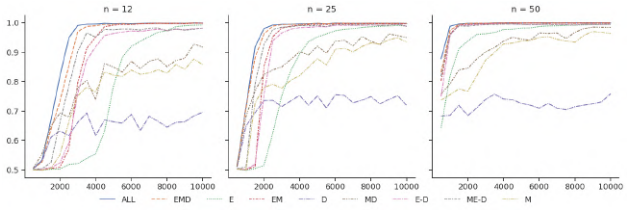


Figure 4. Robustness analysis of the integrated Models of distinct modes with respect to sample size and number of iterations. The horizontal coordinate is the number of epochs and the vertical coordinate is the accuracy.

Table 2 presents the evaluation of semantic segmentation label reconstruction with mIoU scores. Mode ALL, EMD, and MD exhibit comparable performance. Notably, mode MD outperforms mode ME-D, highlighting the significant impact of the decoder component in complex tasks.

Table 2. Evaluation of semantic segmentation label reconstruction with Mean Intersection over Union (mIoU ↑).

ALL	EMD	EM	ME-D	MD	BPS
0.351	0.351	0.350	0.163	0.240	**0.366**

4. Bayesian Power Steering (BPS)

BPS incorporates a pre-trained SD model through the EMD integration mode to deal with task-specific conditions. As an instantiation of the Bayesian formulation in hierarchical levels of the feature space, BPS excels in both computational efficiency and domain recognition across multiple tasks.

4.1. Overview

BPS takes time step information and additional conditions as inputs to perturb the latent features within the pre-trained model, thereby altering the denoising trajectory. Fig.5 shows the main components of BPS: a contracting path (CP), a transition path (TP), and an expansive path (EP). These components are specifically designed to extract features at different scales, enabling adaptation to the diverse hidden features of the pre-trained model.

The dimension of the additional condition $\mathbf{c}_{\text{add}}$ is user-defined, and we take 512×512 for single image-type condition. Following the initial unshuffle module (Shi et al., 2016), we downsample $\mathbf{c}_{\text{add}}$ to 64×64. Subsequently, within the network, the scales for the extracted condition features range from 8×8 to 64×64. Through the components CP, TP, and EP, 8, 1, and 12 condition features are extracted respectively, denoted as $\mathbf{v}_{\text{add}} = \{\mathbf{v}_{\text{add}}^i \mid i = 1, 2, ..., 21\}$.

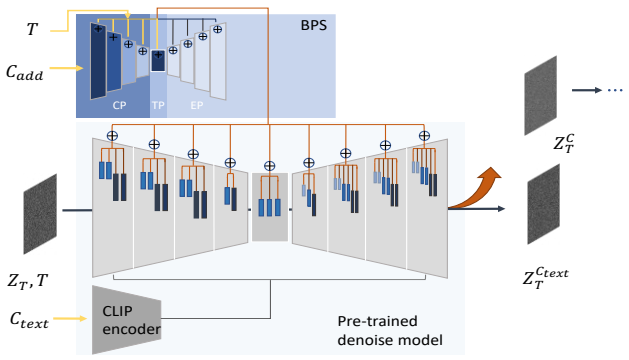


Figure 5. Overview of the integration model. The gray dumbbell shapes represent the pre-trained model. The blue rectangles represent blocks with residual structures, and rectangles of the same color in the same scale represent interventions in the same functional unit sharing features from the BPS.

Note that the SD backbone contains 21 functional units, which refers to the smallest operational unit in the UNet structure and possess residual structures that serve as binding sites. The dimension of $\mathbf{v}_{\text{add}}$ aligns with the dimension of the latent feature $\mathbf{h} = \{\mathbf{h}^{i,j} \mid i = 1, 2, ..., 21, i\text{-th unit}, j\text{-th block}\}$ within the functional unit of the UNet denoiser. The integration of BPS with the blocks across different units is illustrated in Fig.5. The process of feature extraction and fine-tuning latent features at different scales can be summarized as follows:

$$\mathbf{v}_{\text{add}} = B_{\boldsymbol{\phi}}(t, \mathbf{c}_{\text{add}}) \tag{7}$$

$$\hat{\mathbf{h}}^{i,j} = \mathbf{h}^{i,j} + \mathbf{v}_{\text{add}}^i, i = 1, 2, ..., 21, \tag{8}$$

where $B_{\boldsymbol{\phi}}$ represents BPS, which can be considered as the realization of steering gear M defined in (6) resttricted to the feature space, $\boldsymbol{\phi}$ denotes the learnable parameters, and $\hat{\mathbf{h}}$ represents the disturbed latent feature. To train the integration model $\bar{\boldsymbol{\epsilon}}_{\hat{\boldsymbol{\theta}},\boldsymbol{\phi}}(\mathbf{z}, t, \mathbf{c}_{\text{text}}, \mathbf{c}_{\text{add}})$ that combines BPS and the SD backbone, we employ the following optimization process:

$$\mathcal{L}_{\boldsymbol{\phi}} = \mathbb{E}_{Z_0,t,\boldsymbol{\epsilon},\mathbf{c}_{\text{add}}} \left[\left\| \boldsymbol{\epsilon} - \bar{\boldsymbol{\epsilon}}_{\hat{\boldsymbol{\theta}},\boldsymbol{\phi}}(\mathbf{z}, t, \mathbf{c}_{\text{text}}, \mathbf{c}_{\text{add}}) \right\|_2^2 \right]. \tag{9}$$

During the optimization process, the parameter of the pre-trained model $\hat{\boldsymbol{\theta}}$ is frozen, and only the parameter $\boldsymbol{\phi}$ requires updating.

4.2. Architecture Design

Head-heavy and foot-light configuration. BPS mainly consists of various specially designed residual blocks and zero convolution layers. The component CP, TP and EP contains 8, 2 and 12 residual blocks, respectively. Specifically, the residual block structures in CP and TP are identical and consist of two parts. The first part encompasses a convolutional structure responsible for extracting the condition

feature, while the second part combines the temporal features through the scale shift norm mechanism to enable precise temporal control. In contrast, a lightweight residual structure is used in EP, where convolution plays a central role in extracting the condition feature.The detailed structural diagram is provided in Appendix C.2.

Differentiated integration structure. Fig.3 and Table 2 provide insights of the influence by introducing additional information at different stages of the pretrained model. Interventions at earlier stages and middle blocks exhibit a more pronounced effect. With this in mind, basing on the EMD integration mode, we assign different weights to the perturbations originating from different stages. Specifically, for each scale, two residual blocks and one convolution layer are used to extract two latent features from the contracting path. These features are assigned weights $w_i = \frac{\text{scale}}{4}, 1 \leq i \leq 8$, ensuring that earlier stage features have a greater impact. The latent features from TP are assigned a weight of w_1. And the perturbation terms from EP are uniformly assigned a weight of $w_i = 1, 10 \leq i \leq 21$. To this end, the updated formulation becomes:

$$\hat{\mathbf{h}}^{i,j} = \mathbf{h}^{i,j} + w_i * \mathbf{v}^i_{\text{add}}, i = 1, 2, ..., 21. \tag{10}$$

The superiority of such structure design is demonstrated in Table 2.

4.3. Task-Specific Condition Design

The design of conditions (ψ) plays a crucial role in identifying the target domain as demonstrated in Lemma 3.1. It is imperative for BPS to work in concert with the text control layers within the SD backbone. This collaboration is necessary to harness knowledge pertaining to ψ, which in turn facilitates the efficient transmutation of the information contained in C into the image representation Z_0. Moreover, this process must effectively distinguish the signal from the noise disturbance term Z_t to ensure the integrity of the target domain identification. We propose the following condition designs for several tasks.

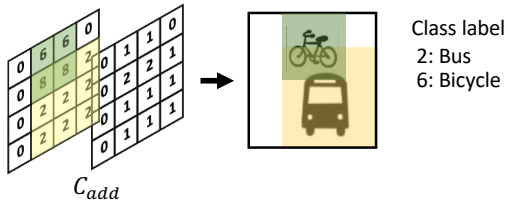


Figure 6. The process of creating a layout condition: perform addition operations on the pixel points of the first channel using layer labels, with the second channel tracking the number of operations on each bit.

Layout condition design. Layout-to-image generation empowers users with precise control over layers. Most existing methods contain complex layer processing and monofunctional neural network architectures (Cheng et al., 2023; Zheng et al., 2023). However, our BPS model effectively handles the task by converting layouts into image-type conditions with dimensions of $512 \times 512 \times 2$. As shown in Fig.6, the first channels capture object location information, while the second channel records object overlap. These conditions, combined with the proposed network structure, contribute to achieving a superior FID of 20.24 on the COCO dataset. More information regarding the experimental setup is provided in Appendix C.1.

Figure 7. Generated samples with layout condition.

Multiple conditions design. Style and line are essential elements in artistic drawing and achieving user-customized artwork. Given the abstract nature of image style, data enhancement techniques and the autoencoder of SD are employed to the latent feature extraction of the source image. The line structure is constructed using the HED soft edge method (Xie & Tu, 2015). In the end, the two conditions are concatenated as input to the BPS, enabling the generation of customized artwork, as shown in Figure 8.

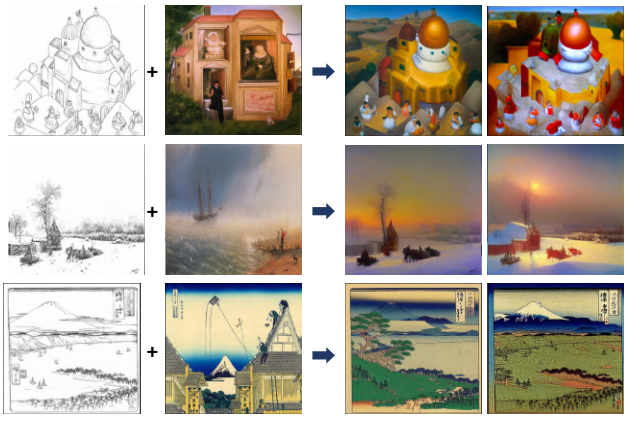


Figure 8. Adapt the pretrained model to a specific art domain with multiple conditions (line, style).

Prompt design. Conditions C_{text} and C_{add} can be highly correlated. To reduce dependence on prompts and enhance

the active capture of information of C_{add}, we design stochastic multilevel textual control. This approach includes three levels of text with equal probability: a generalized overview, such as the default prompt in Zhang et al. (2023) ("a high-quality, detailed, and professional image."), followed by object descriptions (e.g., "a car and a person"), and a detailed portrayal of the objects and their states, generated by BLIP (Li et al., 2022). An example is presented in Fig. 9. These levels of text are incorporated into the integration model during training, with equal probability assigned to each level.

Figure 9. Sketch-to-image. The left showcases the sketch conditions and prompts generated by BLIP, and the right presents the corresponding outputs generated by BPS.

5. Experiments

5.1. Implementation Details

We train multiple BPS models with diverse conditions to customize the pretrained diffusion system for various task-specific scenarios. The pretrained model used is SD V1.5. Training is performed on 2 NVIDIA A100 80GB GPUs, taking at most 48 hours to complete. The optimizer used is AdamW (Loshchilov & Hutter, 2017) with a learning rate of 1×10^{-5}. The effective batch size is 256 after applying gradient accumulation.

The performance of BPS is evaluated in layout-to-Image, artistic drawing, segmentation-to-image, and segmentation-to-image tasks. Further details regarding the datasets are provided in the supplementary materials.

5.2. Qualitative Examination

Figures 7, 8 and 9 showcase the robust performance of BPS across various tasks. Fig. 10 shows the generated images in several prompt settings, demonstrating its ability to accurately interpret content semantics from C_{text} and incorporate both C_{add} and C_{text}.

Figure 10. BPS generated samples in response to prompts regarding time, place, and building materials. Refer to the supplementary material for detailed textual prompts.

5.3. Comparison

We compare our method with T2I-Adapter (Mou et al., 2023), ControlNet (Zhang et al., 2023), and the original SD (Rombach et al., 2022), using the sketch condition in the COCO17 dataset. The dataset consists of 164K images, with the corresponding sketch maps generated by the edge prediction model proposed by Su et al. (2021). T2I-Adapter, ControlNet, and BPS are trained for 10 epochs using the experimental setup in Section 5.1. As shown in Fig. 11, while T2I-Adapter exhibits imaginative capabilities (as indicated by the pink box), both T2I-Adapter and ControlNet exhibit limited attention to details, as evident in the red and yellow boxes. In contrast, BPS demonstrates a refined and hierarchical treatment of details. Furthermore, even in the presence of significant textual errors, BPS demonstrates the ability to achieve relatively unaffected generation quality. However, SD, ControlNet, and T2I-Adapter are noticeably impacted, as highlighted in the blue box.

Table 3. User Preference Ranking for Result Quality and Condition Fidelity. Score 1 to 3 indicates worst to best.

	T2I-Adapter	ControlNet	BPS
Result Quality	1.868	1.77	**2.38**
Condition Fidelity	1.95	1.53	**2.52**

For quantitative evaluation, we employ FID (Seitzer, 2020), CLIP Score (ViT-L/14, Radford et al. (2021)), as presented in Tab.4. We randomly sample 5000, 2500, and 2500 images from the validation set, training set, and testing set, respectively, to obtain the conditions for generation. While there is a slight difference in the CLIP Score due to the impact of the stochastic multilevel textual control we introduced, BPS achieves the best FID score. Furthermore, we conduct the human evaluation following the experimental

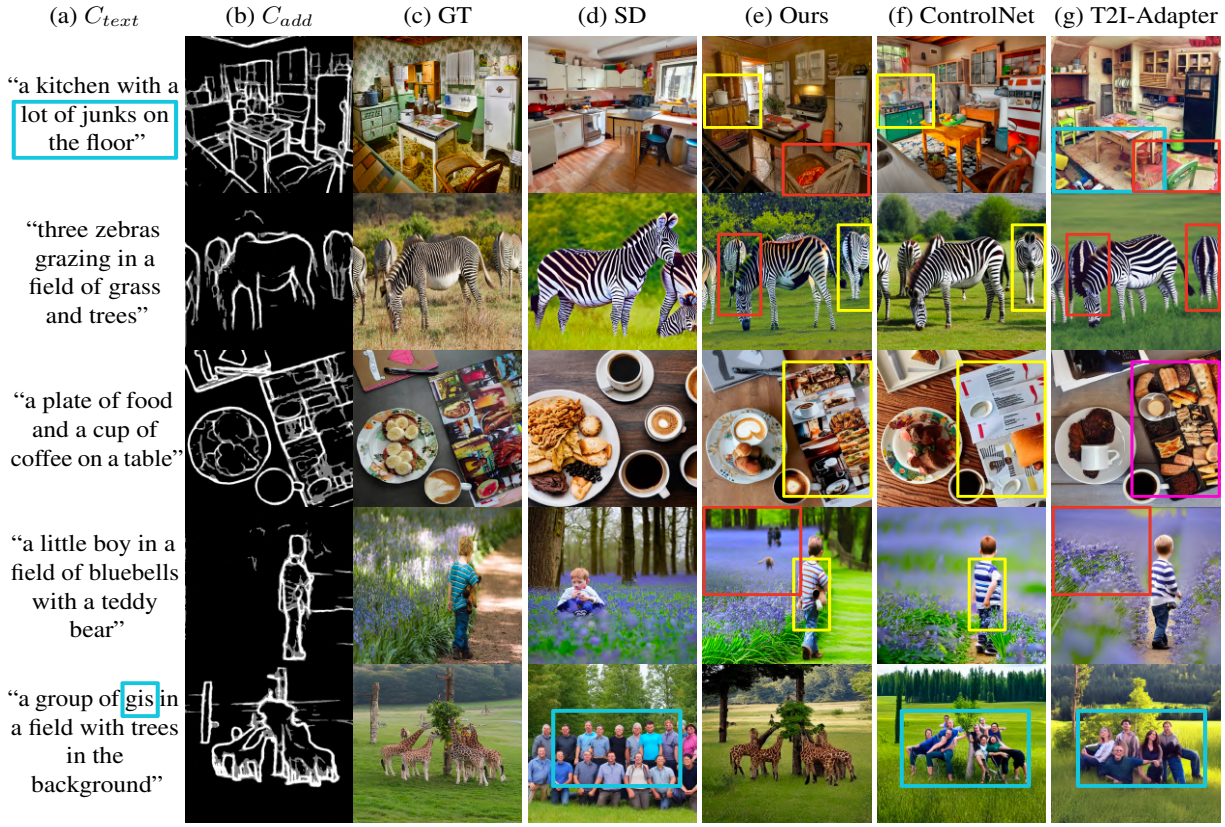


Figure 11. Visualization comparison: our BPS, Stable Diffusion (Rombach et al., 2022), T2I-Adapter (Mou et al., 2023), ControlNet (Zhang et al., 2023), and Ground Truth (GT).

Table 4. Quantitative comparison (FID/CLIP Score). Stable diffusion employs text as the condition, while the other methods utilize text+sketch as the condition.

	SD	T2I-Adapter (Epoch=10)	ControlNet	BPS (Epoch=10)	BPS (Epoch=15)
FID ↓	20.59	18.39	19.41	10.49	**10.04**
CLIP Score ↑	**0.2647**	0.2642	0.2361	0.2614	0.2614

design in Zhang et al. (2023). We sample 25 unseen hand-drawn sketches and assign each sketch to three methods: T2I-Adapter sketch model, ControlNet, and BPS. We invite 20 users to rank these 25 groups of three results based on "the image quality showcased" and "the fidelity to the sketch." User Preference Ranking is used as the preference metric, where users rank each result on a scale of 1 to 3 (lower is worse). As shown in Tab. 3, the results indicate a preference for BPS among the users.

5.4. More Investigation

Additionally, we examine the noteworthy considerations in applications, i.e. time efficiency, data scarcity, and intervention strength.

Implication of time-step information. In the sketch-to-image task, we conduct a controlled experiment by setting the time input of BPS to a constant value, and follow the other setup outlined in Section 5.1. According to Fig.12, the performance declines throughout various training stages when time-step information is absent. It highlights the need to incorporate an additional input, t, into the learning module to enable adaptive adjustment of intervention intensity in the original gradient direction.

Time efficiency. We study time efficiency by training the model up to 15 epochs using the COCO17 dataset. Fig.12 illustrates the FID score of the integrated SD with BPS. It demonstrates satisfactory performance as early as the fourth iteration of training. Moreover, comparing to the initial point, BPS significantly improves the performance of SD.

Data scarcity. To assess the model's generalization ability

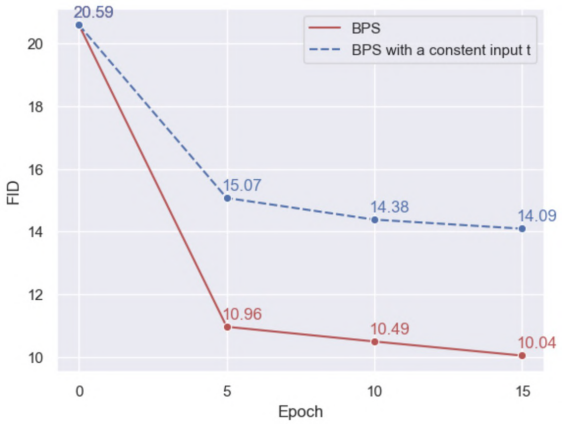


Figure 12. Performance comparison of BPS with constant time input and the original BPS.

in scenarios with limited data, we randomly select subsets of 40, 320, 2562, 20500, and 164K images from the training and validation sets for fine-tuning. Fig.13 presents the fine-tuning performance after 15 iterations using sketches from the test dataset. Our model demonstrates exceptional performance even with a limited number of samples. More results are in the Appendix.

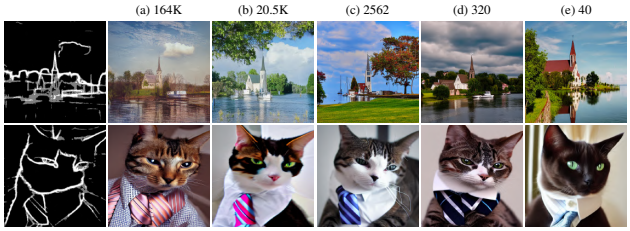


Figure 13. The generation results of models trained with different training set sizes.

Intervention ability. The findings depicted in Fig.14 underscore the significant influence of the intervention weights (w_i, where $1 \leq i \leq 21$) within the BPS framework on the outcomes of the interventions. Specifically, decreasing these weights contributes to an increase in diversity among the generated results.

6. Discussion and Limitations

In this paper, we formulate the problem of generative model domain adaptation and propose a Bayesian fine-tuning framework, which uses the score functions of a pretrained diffusion model as the *prior scores* and updates them to obtain the *posterior scores* using Bayes' theorem. It explores two crucial aspects of domain adaptation: the choice of approximation functions, represented by the neural network structure, and the definition of conditions specific to the task domain. To tackle the first aspect, we introduce a neural

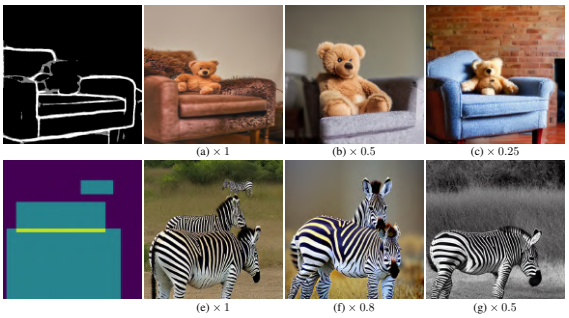


Figure 14. The impact of intervention weights on generated results in layout-to-image and sketch-to-image tasks. The subheading indicates the multiplier applied to the initial weight.

network architecture, Bayesian power steering, for the implementation of the Bayesian formulation in hierarchical levels of the feature space. Our BPS offers several advantages, including (1) exceptional domain recognition and control across different tasks, (2) a compact parameterization and fast convergence, and (3) suitability for data-scarce scenarios with broad applicability. However, our approach still faces certain challenges that warrant further investigation: (a) developing a more refined structure to capture the degree of intervention for multiple conditional inputs, and (b) adaptive adjustment of intervention weights to achieve desired outcomes in extended domain scenarios. These issues remain areas for future study.

Acknowledgment

We extend our sincere appreciation to the Area Chair and the four anonymous reviewers for their time, expertise, and insightful comments. This research was conducted using the computing resources provided by the Research Center for the Mathematical Foundations of Generative AI in the Department of Applied Mathematics at The Hong Kong Polytechnic University. Ting Li's research is partially supported by the National Natural Science Foundation of China (Grant No.12301360) and the research grants from The Hong Kong Polytechnic University. Jian Huang's work is supported by the research grants from The Hong Kong Polytechnic University.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Brack, M., Friedrich, F., Hintersdorf, D., Struppek, L., Schramowski, P., and Kersting, K. Sega: Instructing diffusion using semantic dimensions. *arXiv preprint arXiv:2301.12247*, 2023.

Brooks, T., Holynski, A., and Efros, A. A. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18392–18402, 2023.

Chefer, H., Alaluf, Y., Vinker, Y., Wolf, L., and Cohen-Or, D. Attend-and-excite: Attention-based semantic guidance for text-to-image diffusion models. *ACM Transactions on Graphics (TOG)*, 42(4):1–10, 2023.

Cheng, J., Liang, X., Shi, X., He, T., Xiao, T., and Li, M. Layoutdiffuse: Adapting foundational diffusion models for layout-to-image generation. *arXiv preprint arXiv:2302.08908*, 2023.

Chung, K. L. *A course in probability theory*. Academic press, 2001.

Couairon, G., Verbeek, J., Schwenk, H., and Cord, M. Diffedit: Diffusion-based semantic image editing with mask guidance. *arXiv preprint arXiv:2210.11427*, 2022.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Efron, B. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011.

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

Gafni, O., Polyak, A., Ashual, O., Sheynin, S., Parikh, D., and Taigman, Y. Make-a-scene: Scene-based text-to-image generation with human priors. In *European Conference on Computer Vision*, pp. 89–106. Springer, 2022.

Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A. H., Chechik, G., and Cohen-Or, D. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.

Ghalebikesabi, S., Berrada, L., Gowal, S., Ktena, I., Stanforth, R., Hayes, J., De, S., Smith, S. L., Wiles, O., and Balle, B. Differentially private diffusion models generate useful synthetic images. *arXiv preprint arXiv:2302.13861*, 2023.

Harder, F., Asadabadi, M. J., Sutherland, D. J., and Park, M. Pre-trained perceptual features improve differentially private image generation. *arXiv preprint arXiv:2205.12900*, 2022.

Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., and Cohen-Or, D. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

Ho, J. and Salimans, T. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Ho, J., Saharia, C., Chan, W., Fleet, D. J., Norouzi, M., and Salimans, T. Cascaded diffusion models for high fidelity image generation. *The Journal of Machine Learning Research*, 23(1):2249–2281, 2022.

Hong, S., Lee, G., Jang, W., and Kim, S. Improving sample quality of diffusion models using self-attention guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7462–7471, 2023.

Huang, D., Huang, J., Li, T., and Shen, G. Conditional stochastic interpolation for generative learning. *arXiv preprint arXiv:2312.05579*, 2023.

Jain, J., Li, J., Chiu, M. T., Hassani, A., Orlov, N., and Shi, H. Oneformer: One transformer to rule universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2989–2998, 2023.

Kawar, B., Zada, S., Lang, O., Tov, O., Chang, H., Dekel, T., Mosseri, I., and Irani, M. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6007–6017, 2023.

Kong, Z., Ping, W., Huang, J., Zhao, K., and Catanzaro, B. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020.

Kurakin, A., Ponomareva, N., Syed, U., MacDermed, L., and Terzis, A. Harnessing large-language models to generate private synthetic text. *arXiv preprint arXiv:2306.01684*, 2023.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Li, J., Li, D., Xiong, C., and Hoi, S. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pp. 12888–12900. PMLR, 2022.

Li, K., Gong, C., Li, Z., Zhao, Y., Hou, X., and Wang, T. Meticulously selecting 1% of the dataset for pre-training! generating differentially private images data with semantics query. *arXiv preprint arXiv:2311.12850*, 2023.

Li, X., Tramer, F., Liang, P., and Hashimoto, T. Large language models can be strong differentially private learners. *arXiv preprint arXiv:2110.05679*, 2021.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Luo, S. and Hu, W. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2837–2845, 2021.

Lyu, S., Vinaroz, M., Liu, M. F., and Park, M. Differentially private latent diffusion models. *arXiv preprint arXiv:2305.15759*, 2023.

Mou, C., Wang, X., Xie, L., Zhang, J., Qi, Z., Shan, Y., and Qie, X. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023.

Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

Nichol, A. Q. and Dhariwal, P. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021.

Orgad, H., Kawar, B., and Belinkov, Y. Editing implicit assumptions in text-to-image diffusion models. *arXiv preprint arXiv:2303.08084*, 2023.

Parmar, G., Kumar Singh, K., Zhang, R., Li, Y., Lu, J., and Zhu, J.-Y. Zero-shot image-to-image translation. In *ACM SIGGRAPH 2023 Conference Proceedings*, pp. 1–11, 2023.

Pinaya, W. H., Tudosiu, P.-D., Dafflon, J., Da Costa, P. F., Fernandez, V., Nachev, P., Ourselin, S., and Cardoso, M. J. Brain imaging generation with latent diffusion models. In *MICCAI Workshop on Deep Generative Models*, pp. 117–126. Springer, 2022.

Poole, B., Jain, A., Barron, J. T., and Mildenhall, B. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

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

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp. 234–241. Springer, 2015.

Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., and Aberman, K. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22500–22510, 2023.

Sagers, L. W., Diao, J. A., Melas-Kyriazi, L., Groh, M., Rajpurkar, P., Adamson, A. S., Rotemberg, V., Daneshjou, R., and Manrai, A. K. Augmenting medical image classifiers with synthetic data from latent diffusion models. *arXiv preprint arXiv:2308.12453*, 2023.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35: 36479–36494, 2022.

Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35: 25278–25294, 2022.

Seitzer, M. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid, August 2020. Version 0.3.0.

Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., Bishop, R., Rueckert, D., and Wang, Z. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1874–1883, 2016.

Shi, Y., Wang, P., Ye, J., Long, M., Li, K., and Yang, X. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023.

Sohn, K., Ruiz, N., Lee, K., Chin, D. C., Blok, I., Chang, H., Barber, J., Jiang, L., Entis, G., Li, Y., et al. Styledrop: Text-to-image generation in any style. *arXiv preprint arXiv:2306.00983*, 2023.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020a.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020b.

Su, Z., Liu, W., Yu, Z., Hu, D., Liao, Q., Tian, Q., Pietikäinen, M., and Liu, L. Pixel difference networks for efficient edge detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5117–5127, 2021.

Tan, W. R., Chan, C. S., Aguirre, H. E., and Tanaka, K. Improved artgan for conditional synthesis of natural image and artwork. *IEEE Transactions on Image Processing*, 28(1):394–409, 2018.

Tang, X., Panda, A., Sehwag, V., and Mittal, P. Differentially private image classification by learning priors from random processes. *Advances in Neural Information Processing Systems*, 36, 2024.

Xie, S. and Tu, Z. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 1395–1403, 2015.

Zhang, L., Rao, A., and Agrawala, M. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3836–3847, 2023.

Zheng, G., Zhou, X., Li, X., Qi, Z., Shan, Y., and Li, X. Layoutdiffusion: Controllable diffusion model for layout-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22490–22499, 2023.

Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., and Torralba, A. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 633–641, 2017.

APPENDIX

In the appendix, we provide supplementary theoretical derivations for the BPS introduced in Section 3. Furthermore, we include more detailed description of the implementation of BPS and additional experimental results.

A. Derivation of Equation (5)

Recall that we have

$$\boldsymbol{\epsilon}^*\left(\mathbf{z}_t, t, \mathbf{c}_{\text{text}}\right) := \mathbb{E}\left[\boldsymbol{\eta} \mid \mathbf{z}_t, \mathbf{c}_{\text{text}}\right] = -\sqrt{1-\bar{\alpha}_t}\nabla \log p\left(\mathbf{z}_t \mid \mathbf{c}_{\text{text}}\right)$$

through Tweedie's formula (Efron, 2011), and

$$p(\mathbf{z}_t \mid \mathbf{c}_{\text{text}}, \mathbf{c}_{\text{add}}) = \frac{p(\mathbf{c}_{\text{add}} \mid \mathbf{c}_{\text{text}}, \mathbf{z}_t) p(\mathbf{z}_t \mid \mathbf{c}_{\text{text}})}{p(\mathbf{c}_{\text{add}} \mid \mathbf{c}_{\text{text}})} \tag{11}$$

by Bayes' theorem.

Take the logarithm of both sides of Equation (11) and subsequently compute the gradient with respect to $\mathbf{z}_t$, then we obtain

$$\begin{aligned}\nabla \log p(\mathbf{z}_t \mid \mathbf{c}_{\text{text}}, \mathbf{c}_{\text{add}}) &= \nabla \log p(\mathbf{c}_{\text{add}} \mid \mathbf{c}_{\text{text}}, \mathbf{z}_t) + \nabla \log p(\mathbf{z}_t \mid \mathbf{c}_{\text{text}}) - \nabla \log p(\mathbf{c}_{\text{add}} \mid \mathbf{c}_{\text{text}}) \\ &= \nabla \log p(\mathbf{c}_{\text{add}} \mid \mathbf{c}_{\text{text}}, \mathbf{z}_t) + \nabla \log p(\mathbf{z}_t \mid \mathbf{c}_{\text{text}}),\end{aligned}$$

where $p(\mathbf{c}_{\text{add}} \mid \mathbf{c}_{\text{text}})$ is constant for $\mathbf{z}_t$ so that $\nabla \log p(\mathbf{c}_{\text{add}} \mid \mathbf{c}_{\text{text}}) = 0$.

Combining the above formulas, the integrated denoise function can be derived as follows.

$$\begin{aligned}\bar{\boldsymbol{\epsilon}}^*(\mathbf{z}_t, t, \mathbf{c}_{\text{text}}, \mathbf{c}_{\text{add}}) := \mathbb{E}\left[\boldsymbol{\eta} \mid \mathbf{z}_t, \mathbf{c}_{\text{text}}, \mathbf{c}_{\text{add}}\right] &= -\sqrt{1-\bar{\alpha}_t}\nabla \log p(\mathbf{z}_t \mid \mathbf{c}_{\text{text}}, \mathbf{c}_{\text{add}}) \\ &= -\sqrt{1-\bar{\alpha}_t}\left[\nabla \log p(\mathbf{c}_{\text{add}} \mid \mathbf{z}_t, \mathbf{c}_{\text{text}}) + \nabla \log p(\mathbf{z}_t \mid \mathbf{c}_{\text{text}})\right] \\ &= -\sqrt{1-\bar{\alpha}_t}\nabla \log p(\mathbf{c}_{\text{add}} \mid \mathbf{z}_t, \mathbf{c}_{\text{text}}) + \boldsymbol{\epsilon}^*(\mathbf{z}_t, t, \mathbf{c}_{\text{text}}),\end{aligned}$$

which complete the derivation.

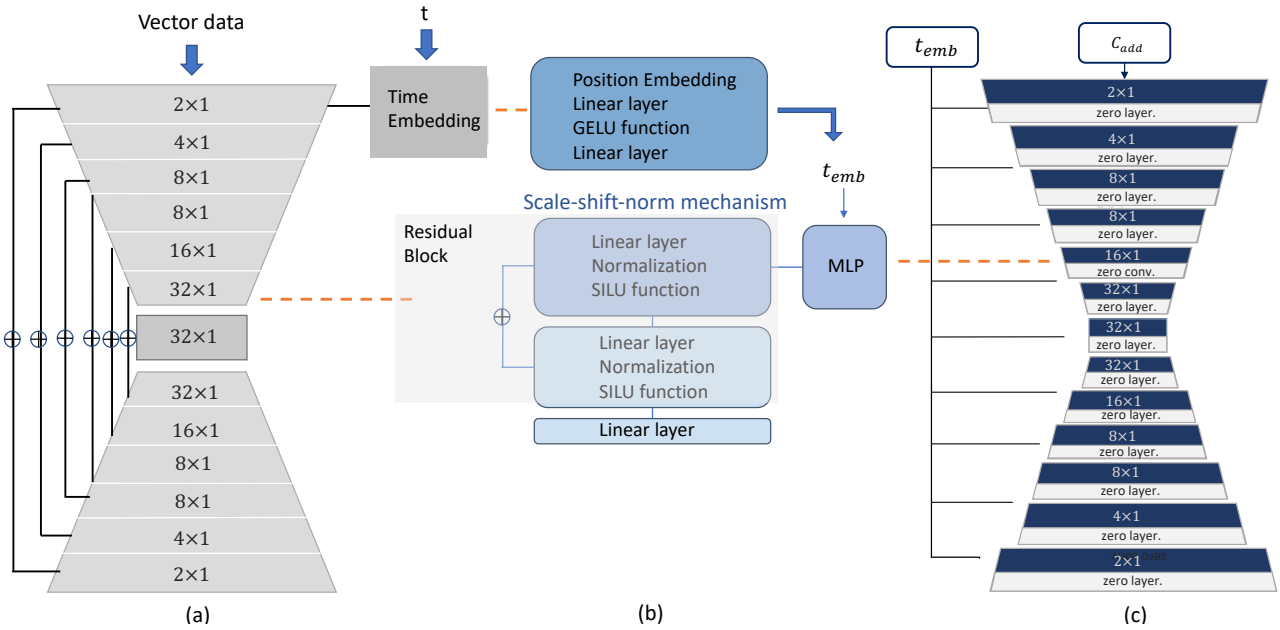


Figure 15. Architecture of pre-trained model and learnable modules for vector data.

B. Generation of 2D Data

In this setting, the samples exhibit a multimodal distribution, with a probability of 0.7 uniformly distributed on the first ring and 0.3 on the second ring. The centers of the rings are positioned at coordinates $(0, 0)$ and $(5, 0)$ in the 2D coordinate space. Both rings have identical sizes and feature inner and outer radii of 0.6 and 1 respectively.

During the pretraining stage, we randomly sampled 500K samples from the rings to train the prior denoise model. In the fine-tuning stage, only 100 samples with support set information (i.e., labels indicating the originating ring) were available for training. The objective is to generate samples adhering to the specified ring distribution.

B.1. Network Structure

We develop a U-net architecture specifically designed for vector data. This pre-trained U-Net model, shown in Fig. 15(a), serves as the backbone for fine-tuning. Data features are scaled into 2, 4, 8, 16, and 32 dimensions across blocks, each comprising a Residual block and a linear layer, as depicted in Fig. 15(b).

The fine-tuning process involves learnable modules following the structure illustrated in Fig. 15(c). Notably, these modules incorporate a "zero layer" - a fully connected layer initialized with zero weights. The intermediate outputs from the learnable module are first processed through the zero layer before being combined with the outputs of the pre-trained model. This ensures a smooth evolution of the original signals (i.e., the outputs of the pre-trained model) during the training process, facilitating model convergence.

B.2. Quantitative Evaluation

We evaluate distinct integration modes to fine-tune the pre-trained model. After generating samples using the ground truth labels and the integrated model, we employ a linear binary classifier to determine the model to which the samples belong. The computed accuracy and precision metrics are then used to assess the efficacy of the integrated model configurations.

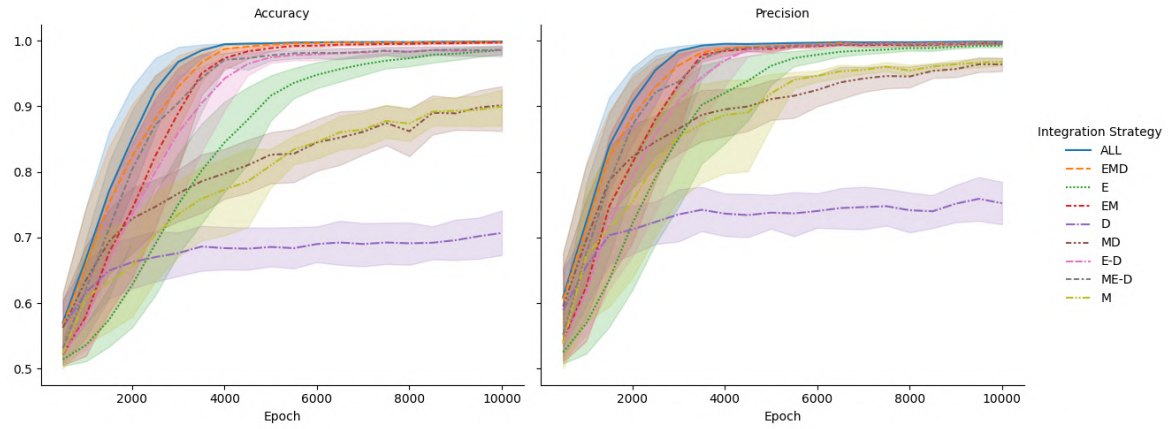


Figure 16. Accuracy and precision of model classification using generated samples from aforementioned integration modes at different training stages with fine-tuning sample size $n = 12$. Five groups of samples are randomly selected for experimentation across different modes.

As shown in Fig. 16, the integrated modes ALL, EMD, and EM achieve the best performance in terms of both metrics and exhibit the fastest convergence rates. Although extended training allows modes ME-D, E-D, and E to approach similar precision levels, they still trail the performance of modes ALL, EMD, and EM in terms of accuracy, even with prolonged training. In contrast, the models under modes M, MD, and D exhibit the poorest overall performance. These results preliminarily suggest that interventions on the early intermediate outputs of the pre-trained model are crucial for optimal fine-tuning.

Furthermore, to model realistic scenarios with limited data resources, fine-tuning is performed using a small number of labeled samples, ranging from 12 to 400. In this experiment, we use accuracy as the evaluation metric to validate the performance of the different integrated models. This approach allows us to assess the robustness of these integration models under resource-constrained settings.

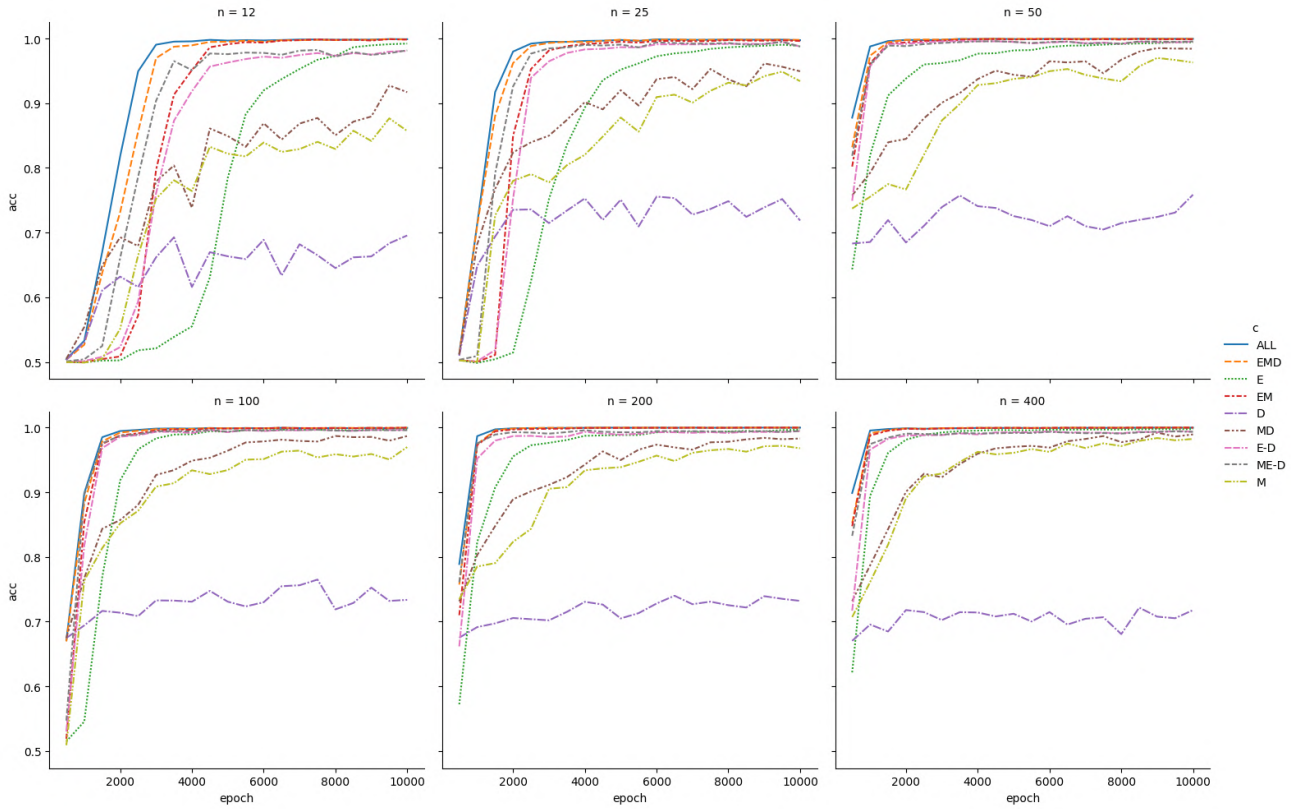


Figure 17. Accuracy of model classification using generated samples from distinct integration modes across data sample sizes.

Notably, even as the training sample size decreases, the integrated modes ALL, EMD, and EM exhibit the highest robustness, with their performance accuracy remaining relatively stable at the peak level. In contrast, the other modes experience varying degrees of decline in convergence speed and overall performance, with models under modes M and M-D showing the most significant drops. And modes ME-D and E-D demonstrate relative instability under the most data-constrained conditions (n=12, 25). This trend is more clearly illustrated in the comparison of different models at the same training epoch, as shown in the figure below.

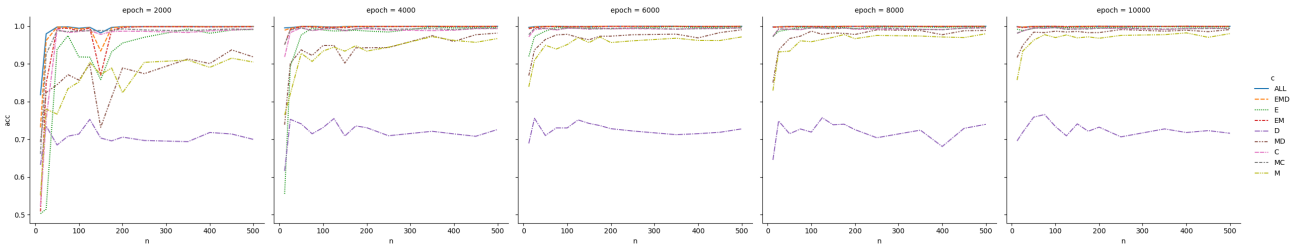


Figure 18. Accuracy of model classification using generated samples from distinct integration modes across training stages.

C. Image Generation

C.1. Training Parameters

We train multiple BPS models for various tasks, as detailed below.

Segmentation-to-image. For this structure condition, we utilize ADE20K (Zhou et al., 2017), consisting of 25,574 images, as the training dataset. Its semantic segmentation encompasses a remarkable 193,238 annotated object parts, including

intricate sub-parts. The BPS model is trained using 2 NVIDIA A100 80GB GPUs, with the Stable Diffusion V1.5 as the base model. The batch size is set to 256, achieved through a physical batch size of 16 and 4× gradient accumulation. We do not use ema weights.

Layout-to-Image. We evaluate the performance of bounding box layout-to-image using the COCO17 dataset (Lin et al., 2014). Data preprocessing follows the approach outlined in Cheng et al. (2023), where we filter images to contain a range of 3 to 8 objects. Additionally, objects occupying less than 2% of the image area are excluded. Consequently, our training dataset comprises 112,680 samples, and the validation dataset contains 3,097 samples. The BPS model is trained using 2 NVIDIA A100 80GB GPUs, with the Stable Diffusion V1.5 as the base model. The effective batch size is 128 after applying gradient accumulation. We do not use ema weights.

Artistic drawing. To train our model for this application, we utilize the WikiArt Dataset (Tan et al., 2018). The dataset was carefully selected to include works from painters with more consistent styles, resulting in a training set of 6,285 samples from 17 artists and 12 styles. The artists included in the dataset are Camille Pissarro, Childe Hassam, Claude Monet, Edgar Degas, Eugene Boudin, Gustave Dore, Ivan Aivazovsky, Marc Chagall, Pablo Picasso, Paul Cezanne, Pierre Auguste Renoir, Raphael Kirchner, Fernando Botero, Bernardo Bellotto, Cornelis Springer, Katsushika Hokusai, and Hiroshige. The art styles represented in the dataset span various genres, including Impressionism, Realism, Pointillism, Romanticism, Naive Art Primitivism, Cubism, Analytical Cubism, Synthetic Cubism, Post Impressionism, Art Nouveau Modern, Rococo, and Ukiyo-e. We train the BPS model on 2 NVIDIA A100 80GB GPUs. The batch size is set to 512, achieved through a physical batch size of 32 and 16× gradient accumulation. We do not use ema weights.

Sketch-to-image. For this application, we leverage the COCO17 dataset (Lin et al., 2014) as our training data, comprising a rich collection of 164,000 images. To generate the corresponding sketch maps, we employ the edge prediction model proposed by (Su et al., 2021). The BPS model is trained using 4 NVIDIA A100 80GB GPUs, with the Stable Diffusion V1.5 as the base model. The batch size is set to 256, achieved through a physical batch size of 32 and 8× gradient accumulation. We do not use ema weights.

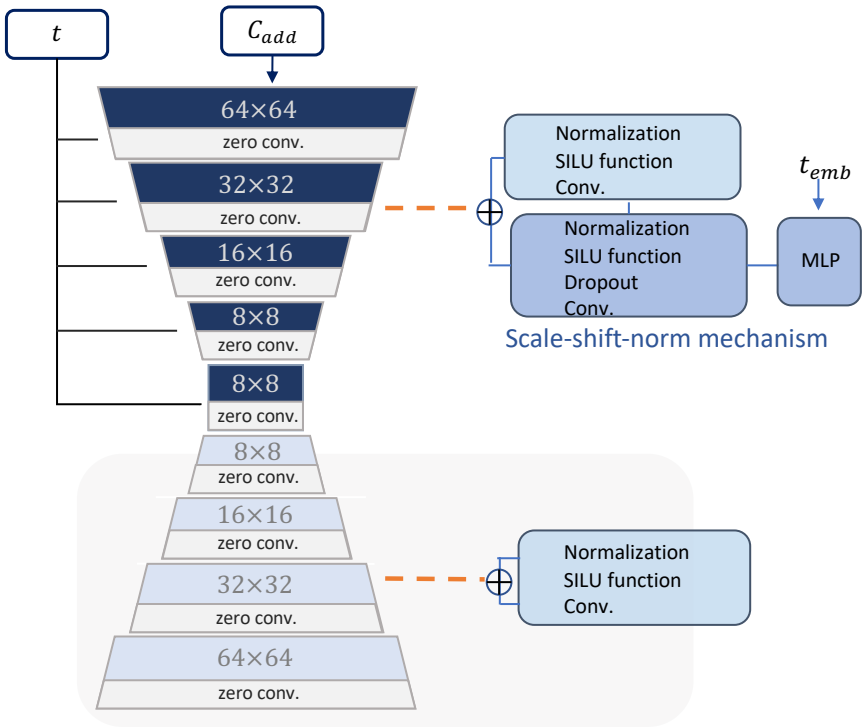


Figure 19. Architecture of BPS.

C.2. Network Structure: BPS

The BPS network structure employs a top-heavy design, as depicted in Fig 19. The dark block incorporates a residual block that primarily utilizes a Scale-shift-norm mechanism to fuse temporal and conditional information. On the other hand,

the light block consists of a layer of residual blocks. Notably, the blocks incorporate several "zero conv." - convolution layers initialized with zero weights. The intermediate outputs from the learnable module are first processed through the zero convolution layer before being combined with the outputs of the pre-trained model. This ensures a smooth evolution of the original signals (i.e., the outputs of the pre-trained model) during the training process, facilitating model convergence (Zhang et al., 2023).

C.3. Inference Settings

Sampler: we employ Denoising Diffusion Implicit Model (DDIM) as the sampler for our approach. We utilize 50 steps to sample each image.

Prompts: the segmentation-to-image and Layout-to-Image tasks utilize object descriptions as text prompts (e.g., "oven, microwave, book"). For the Artistic Drawing task, the description includes both the subject matter and the artist (e.g., "chestnut trees at Louveciennes from Camille Pissarro's brush"). Lastly, the Sketch-to-image task adopts prompts generated by the BLIP method (Li et al., 2022).

C.4. Quantitative evaluation

Segmentation-to-image. The performance of the BPS model is evaluated on the ADE20K dataset (Zhou et al., 2017) using the Mean Intersection-over-Union (mIoU) metric to assess conditioning fidelity. The state-of-the-art segmentation method, OneFormer (Jain et al., 2023), achieves an mIoU of 0.58 on this dataset. We generate images using segmentations from the ADE20K validation set and then feed the generative results to OneFormer for segmentation detection and computation of reconstructed IoUs. The BPS model achieve a score of 0.366, outperforming the ControlNet approach (Zhang et al., 2023), which reported a score of 0.35.

Layout-to-Image. The generated images are evaluated at a resolution of 256×256. We adopt the Fréchet Inception Distance (FID) metric, as used in (Cheng et al., 2023), to assess the performance of the BPS model on this task. After training for 30 and 60 epochs, the BPS model achieves FID scores of 20.47 and 20.24, respectively, which are comparable to the state-of-the-art score of 20.27 reported in (Cheng et al., 2023). This indicates that the BPS model, even with a relatively simple network structure, can achieve results on par with the current state-of-the-art.

Sketch-to-image. We employ FID (Seitzer, 2020), CLIP Score (ViT-L/14, Radford et al. (2021)), as presented in Tab. 4. We randomly sample 5000, 2500, and 2500 images from the validation set, training set, and testing set, respectively, to obtain the sketch conditions for generation. While there is a slight difference in the CLIP Score due to the impact of the stochastic multilevel textual control we introduced, BPS achieve the best FID score, surpassing the competitio.

C.5. Qualitative Evaluation

The BPS model does not consider C_{text} as an input. Investigating the integration of the BPS model with diverse textual prompts is necessary. The following example demonstrates the effective integration of inputs derived from various textual sources with the BPS model. Detailed textual descriptions are shown in Fig. C.5.

(a) C_{add}

(b) “Nestled amidst lush foliage, the country house unveils its enchanting beauty as summer cascades upon it.”

(c) “Amidst the fall’s magical embrace, a country house stands adorned with a vibrant carpet of fallen leaves.”

(d) “On a serene winter day, the country house stands gracefully amidst a pristine blanket of snow. ”

(e) “A wooden house built in the desert.”

(f) “a house is built under the sea.”

(g) “” a country house is constructed with toy bricks.”

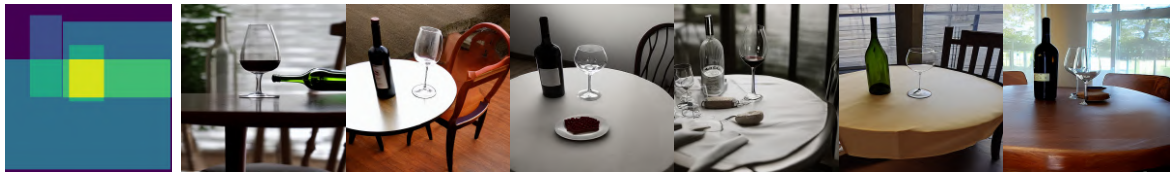
(h) “a House made of toy plasticine.”

Furthermore, regarding the generation tasks discussed in the paper, we provide the following generation samples. These experimental results demonstrate the BPS model’s powerful generative capabilities and strong conditioning fidelity.

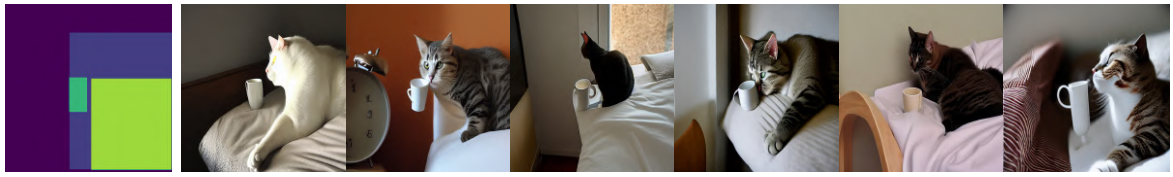
Figure 20. Segmentation-to-image. The first row is the condition, and the rest of the rows in each column are the corresponding generated structures

“train”

“oven, microwave, book”



“bottle, chair, dining table, wine glass”



“cat, bed, cup”

Figure 21. Layout-to-Image. The left column is the condition and the right column is the generated result.

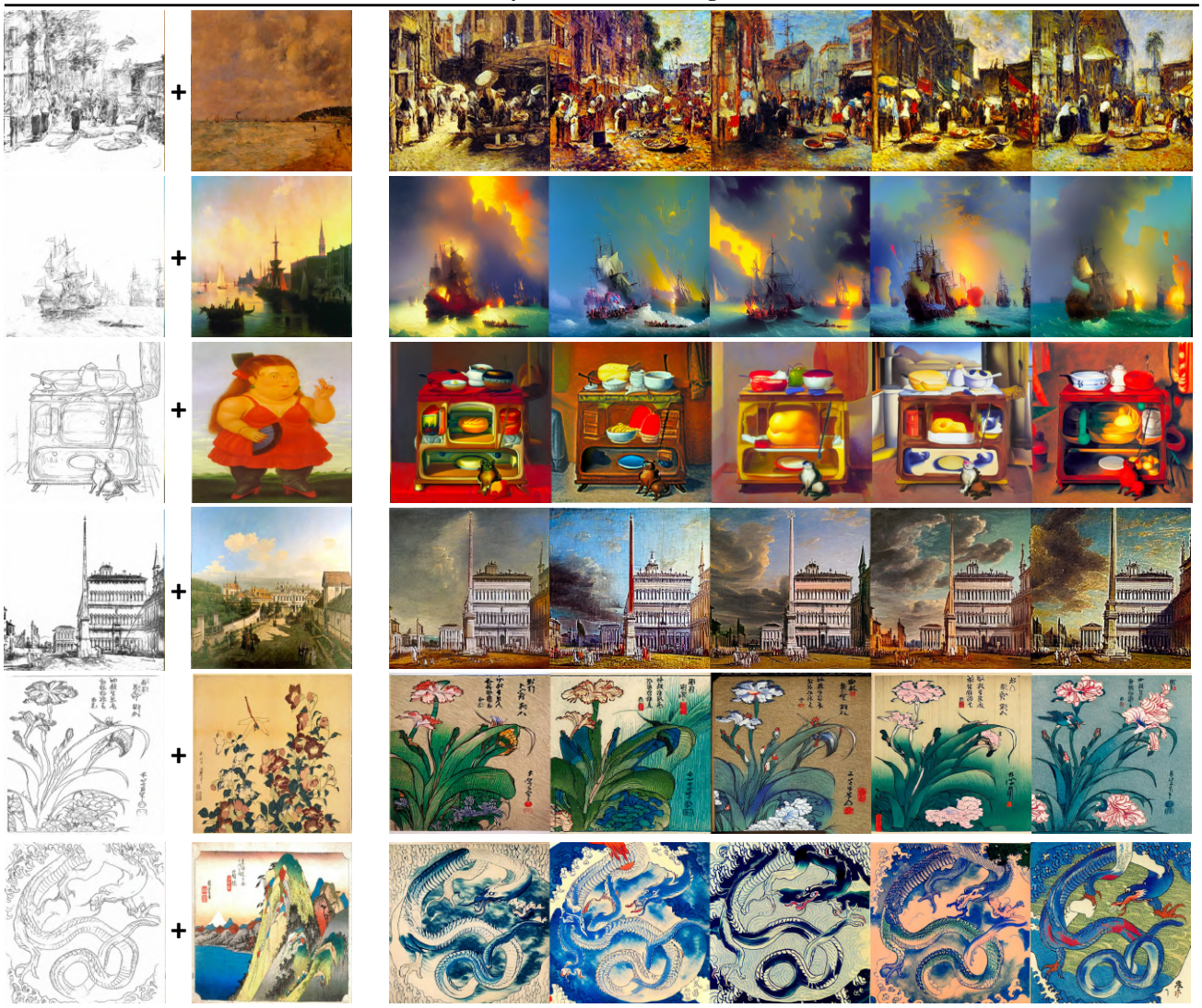


Figure 22. Artistic drawing. The left column is the line condition and the source image, the right column is the generated result.

Figure 23. Sketch-to-image. The first row is the condition, and the rest of the rows in each column are the corresponding generated structures.

D. Comparison

Fig. 24 presents a comparative analysis of the various methodologies employed in this study. The text data, generated using BLIP (Li et al., 2022), aligns coherently with the main context of the research.

Figure 24. Comparison between our BPS, ControlNet (Zhang et al., 2023), T2I-Adapter (Mou et al., 2023).

E. Ablation Study

E.1. Data Efficiency

This study utilized the coco17 dataset, with a random selection of data from the training and validation sets for model training. The models were then tested on the separate test set, and the generated results are presented below. Columns (a)-(e) in Fig. 25 provide a comparative analysis of the generation results obtained from models trained with datasets of different sizes.

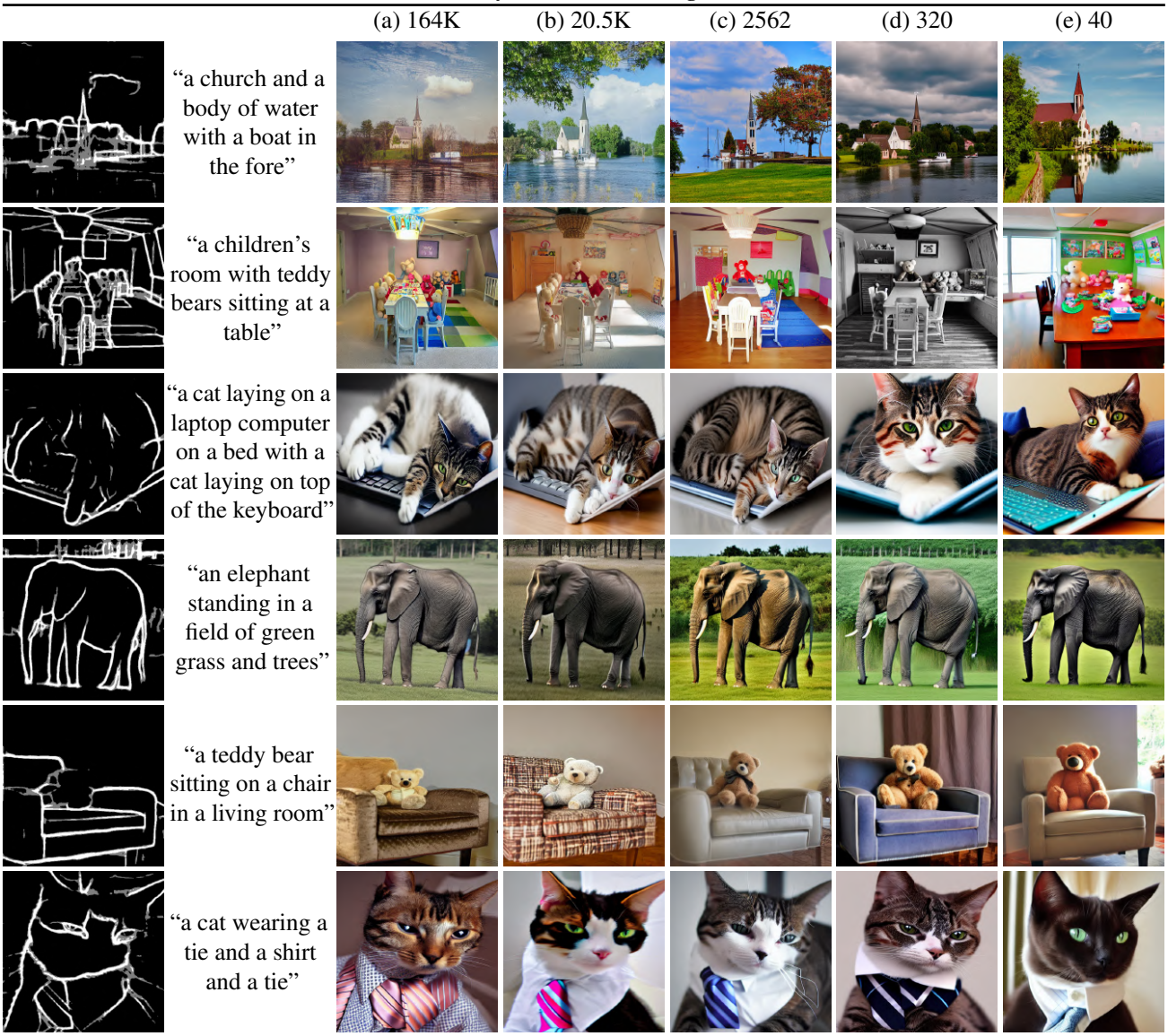


Figure 25. Comparative analysis of the generation results achieved by models trained using datasets of varying sizes.

E.2. Intervention Strength

This study examined the impact of varying intervention weights on control effects in two distinct tasks: the layout-to-image task and the sketch-to-image task.

Layout-to-image. In this experiment, the weights $w_i, i = 1, ..., 21$ were uniformly initialized to 1. The effects generated under different weight settings, specifically 1, 0.8, and 0.5, are illustrated in columns (a), (b), and (c) of the figure below.

Sketch-to-image. In this experiment, the weights were initialized as $w_i = [16, 16, 8, 8, 4, 4, 2, 2, 16]$ for $i = 1, ..., 9$, and $w_i = 1$ for $i = 10, ..., 21$, in line with the configuration described in subsection 4.2. Columns (a), (b), and (c) in Fig. E.2 demonstrate the down-weighting of intervention weights for the encoder and middle block to 1, 1/2, and 1/4, respectively.

(a) Intervention weight settings and their impact on generation effects in the Layout-to-image task are depicted in the figure. Columns (a), (b), and (c) represent different intervention weights applied to the same BPS model.

(b) Intervention weight settings and their impact on generation effects in the sketch-to-image task are depicted in the figure. Columns (a), (b), and (c) represent different intervention weights applied to the same BPS model.