RISSOLE: Parameter-efficient Diffusion Models via Block-wise Generation and Retrieval-Guidance

Avideep Mukherjee[1]
https://www.cse.iitk.ac.in/users/avideep/

Soumya Banerjee[1]
https://soubanerjee.github.io/

Piyush Rai[1]
https://www.cse.iitk.ac.in/users/piyush/

Vinay P. Namboodiri[2]
https://vinaypn.github.io/

[1] Department of Computer Science and Engineering
Indian Institute of Technology Kanpur
Uttar Pradesh, India

[2] Department of Computer Science
University of Bath
Claverton Down
Bath BA2 7AY, UK

Abstract

Diffusion-based models demonstrate impressive generation capabilities. However, they also have a massive number of parameters, resulting in enormous model sizes, thus making them unsuitable for deployment on resource-constraint devices. An approach based on block-wise generation holds considerable promise toward designing compact-sized (parameter-efficient) deep generative models since the model generates only a block instead of the whole image at once. However, block-wise generation is also considerably challenging, requiring ensuring coherence across the generated blocks. We design a retrieval-augmented generation (RAG) approach and leverage the corresponding blocks of the images retrieved by the RAG module to condition the training and generation stages of a block-wise denoising diffusion model. Our conditioning schemes ensure coherence across the different blocks during training and, consequently, during generation. While we showcase our approach using the latent diffusion model (LDM) as the base model, it can be used with other variants of denoising diffusion models. We validate the solution of the coherence problem through the proposed approach by reporting substantive experiments to demonstrate our approach's effectiveness in compact model size and excellent generation quality.

1 Introduction

Recent work on deep generative models, particularly diffusion probabilistic models and variants [10, 18, 20], has produced impressive generation quality for data such as images and videos. While recent models can now generate images with very high resolutions, these models also have enormous sizes, which hinders their deployment in resource-constrained settings. Training a state-of-the-art diffusion model from scratch could, for instance, require 23,835 A100 hours (when optimized). [1] Clearly, these models are beyond the computational

---

[1]An example provided in https://www.databricks.com/blog/stable-diffusion-2

capabilities of academia or organizations with modest computing resources, which necessitates a parameter-efficient design. To address the issue of computationally intensive training and large model sizes, recent work has explored approaches, such as training denoising diffusion models in a lower-dimensional latent space [14], and using *semi-parametric* retrieval-augmented diffusion models [2] where a compact-sized diffusion model is conditioned on an external database. As the training of the diffusion model is done with a fixed latent size, these models still tend to work with large latent representations, resulting in large parameter models that are challenging to train. *Post hoc* attempts at optimization such as adversarial diffusion distillation [17] are also not applicable as one still needs access to a fully trained large parameter model and further resources to train a distilled model.

Training deep generative models such that they can generate images in a *block-wise* fashion (generating one block at a time), instead of the whole image at once, can potentially be an alternative to address the aforementioned issues. However, designing such block-wise generative models is quite challenging, as ensuring that the generated blocks have spatial and semantic coherence is difficult. Consequently, the generated blocks may not blend well together, resulting in noticeable discrepancies or artifacts where the blocks meet.

In this work, we show that the retrieval-augmented generation (RAG) approach [2] can effectively address the consistency problem in block-wise generation. In particular, we present a block-wise denoising diffusion model where each image block (in the pixel space, or the latent space if using a latent diffusion model [14]), both during training as well as generation, is conditioned on the corresponding block of a reference set of images retrieved from an external database. We show that enforcing correspondence in the block-wise conditioning suffices to obtain coherence in the *independent* block-wise generation.

2 Related Work

There is extensive work on deep generative models for data, such as images, videos, text, and multi-modal data. Here, we mainly focus on generative models based on diffusion probabilistic models [18] and their variants [5, 10, 11, 15, 16, 19, 20], particularly those focusing on compact model sizes, as they are most relevant to our work. There has been an extensive exploration of variants of diffusion-based models, such as score-based models and their application on various data sources [24].

Our work mainly focuses on designing diffusion models that are parameter-efficient *by design* and can consequently be trained efficiently. There are three main strategies: optimizing the representation (using probability flow ODEs), training using a latent representation, and patch-based training. Models such as consistency models [21] obtain an approximation for the stochastic differential equation based on the diffusion process. However, the main contribution of this approach is to reduce the number of function evaluations required at inference time. This approach still requires substantial training time. Methods such as the latent diffusion model (LDM) [14] model the diffusion process in the latent space instead of the original image space. One could theoretically use a highly compact latent space to reduce the computational cost. However, as the efficacy of the encoder-decoder transformation is limited, typically, one uses a sizeable latent space representation while training these models, and these LDMs require thousands of hours of large GPU-based training.

Patch-based training of diffusion models is an alternative. For instance, [22] shows that training on patches could reduce the training time. However, they do not solve the coherence issue. Their strategy is to train with various patches of various sizes and to train using a few *full-sized* images during training, thus not yielding any model-sized improvements.

Another recent patch-based strategy [2] trains with a patch-based model. However, they provide the context from the feature collage of patches to appropriately condition the image generation. We do evaluate a variant of this approach where we condition on adjacent blocks. We observed that the representation does not train well with our proposed strategy.

Our work is based on more explicitly and directly incorporating coherence by conditioning the individual blocks on the corresponding blocks of "relevant" images retrieved from an external database using the idea of retrieval-based augmentation. Our work in incorporating coherence also has relevance to the broader area of coherence in generative models [3, 23]. Our proposed approach is to develop block-based retrieval-guided diffusion models. This is complementary to other approaches of probability flow ODE [21] or latent diffusion [14] and can be combined with these different approaches.

3 Background

Diffusion Models: Denoising diffusion probabilistic models (DDPM) [10, 18, 20] are deep generative models that consist of a forward process and a reverse process. The forward process $q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1})$ takes a clean input x_0 and gradually corrupts it by producing a sequence $x_1, x_2, \ldots, x_T$ defined by forward diffusion of the form $q(x_t|x_{t-1})$, usually modeled by a Gaussian $q(x_t|x_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}x_t, \beta_t\mathbf{I})$. Note that $q(x_t|x_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I})$, where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_t \alpha_t$. For a sufficiently large value of T and a suitably chosen variance schedule $\{\beta_t\}_{t=1}^{T}$, the distribution $q(x_T|x_0)$ approximates an isotropic Gaussian. A *learnable* reverse process, called reverse denoising, denoted as $p(x_{0:T}) = p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$, tries to reconstruct the clean data x_0 from pure noise x_T with $p(x_T) = \mathcal{N}(0, \mathbf{I})$ and $p_\theta(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$, where the parameters $\mu_\theta(x_t, t), \Sigma_\theta(x_t, t)$ of the denoising network are defined using a deep neural network. Once learned, the reverse process can synthesize new inputs from pure noise x_T. It can also be shown that estimating the reverse process parameters is equivalent to predicting the noise in the distribution $q(x_t|x_0)$.

Latent Diffusion Models: While the forward and reverse diffusion processes of the standard DDPM operate in the image space, the latent diffusion model (LDM) [14] takes a different approach where both of these processes operate in a low-dimensional latent space. The LDM consists of an encoder $\mathcal{E}$ which compresses the input image x into a lower-dimensional latent representation z, a decoder $\mathcal{D}$ that reconstructs x from z, and the forward and reverse diffusion processes operate on the latent z representation. In addition to yielding a semantically meaningful latent space, LDM has immediate benefits regarding reduced model size, training cost, and sample generation speed.

Retrieval-Augmented Generation: An orthogonal approach to designing compact-sized deep generative models is retrieval-augmented generation (RAG) [25] which augments a compact generative model with an external database $\mathcal{D}$. RAG uses similarity-based retrieval during training/generation to condition the generative model based on additional data retrieved from this external database. Blatmann et al. [2] leveraged the RAG idea for diffusion models by employing a non-parametric retrieval strategy $\xi_k : \mathcal{D} \mapsto \mathcal{M}_{\mathcal{D}}^{(k)}$ such that $\mathcal{M}_{\mathcal{D}}^{(k)} \subseteq \mathcal{D}$ and $|\mathcal{M}_{\mathcal{D}}^{(k)}| = k$. During the training phase, for each sample $x \sim p(x)$, their proposed retrieval-augmented diffusion model (RDM) uses ξ_k to acquire a set of k nearest neighbors $\mathcal{M}_{\mathcal{D}}^{(k)}(x)$ from the database $\mathcal{D}$. Subsequently, the generative model is conditioned on the representations of these retrieved instances, eliminating the necessity to synthesize an image region entirely from its parameter manifold. While RAG is primarily motivated by the need for

compact models, our work shows how RAG can also be leveraged to solve the challenging problem of ensuring coherence across blocks when doing block-wise generation.

4 Denoising Diffusion Models with Block-wise Generation

Our block-wise generation approach applies to DDPMs operating in the image space and those operating in a latent space. This work will illustrate our approach using the latent diffusion model (LDM) [14], an example of the latter class of methods. Our approach (Fig. 1) assumes that the input to each forward/reverse diffusion step is partitioned into b disjoint blocks. Omitting the time-step subscript t and denoting the latent representation at any time-step as z, we assume it to be partitioned into b equal-sized disjoint blocks as $z = \bigcup_{i=0}^{b-1} z^i$. Operating on smaller-sized blocks z^i naturally results in a model with fewer parameters, which is our primary goal. However, the challenge here is to ensure coherence among the blocks (e.g., avoiding artifacts at the block boundaries, semantic inconsistency between adjacent blocks, etc.). To achieve coherence, each block z^i is also conditioned on the *corresponding* block of a set of k nearest neighbors retrieved from the external database; Sec. 4.1.3 provides the details. We refer to our approach as RISSOLE (ParameteR-efficIent DiffuSion Models via Block-wiSe GeneratiOn and RetrievaL-GuidancE).

4.1 RISSOLE Training

The training procedure of RISSOLE consists of the following steps: (1) VQ-GAN encoder-decoder training because our model operates in a latent space,(2) Building the database from which retrieval is performed, and (3) DDPM training in the latent space using a block-wise generation approach and leveraging RAG to ensure coherence across blocks. Next, we describe each of these steps in more detail.

4.1.1 VQ-GAN Training

As the first stage of RISSOLE training, we train a VQ-GAN on the original training images to obtain their latent representations using a trainable encoder-decoder setup. For an image $x \in \mathcal{X} \subset \mathbb{R}^{h \times w \times c}$, we introduce an encoder E_θ, where E_θ takes x as input and produces a hidden representation $z \in \mathbb{R}^{h' \times w' \times c'}$. Here, h, w, c represents the dimensions of the input image, h', w', c' represents the dimensions of the hidden representation, and θ represents the parameters of the encoder E s.t. $z = E_\theta(x)$. Similar to LDM [14], and to be amenable to our subsequent block division approach, we maintain z in a two-dimensional layout, departing from the arbitrary one-dimensional arrangement used in the works of [8] and [13]. This reduces the image space by a factor of $f = w/w'$ or h/h'. The factor f determines the size of each block into which the hidden representations are divided. A decoder D_ϕ, parameterized by ϕ, reconstructs the original image as $\tilde{x} = D_\phi(E_\theta(x)) \in \mathbb{R}^{h \times w \times c}$.

4.1.2 Building the Retrieval Database

For constructing the retrieval database, denoted as $\mathcal{D}$, the latent representations z of the training images are partitioned into b non-overlapping, equal-sized blocks. We denote $\mathcal{D}$ as $\bigcup_{i=0}^{b-1} \mathcal{D}_i$, where $\mathcal{D}_i$ comprises a stack of the ith blocks of $z = E(x)$ for all $x \in \mathcal{X}$.

4.1.3 Block-wise DDPM Training

Recall that the forward and reverse diffusion of our proposed DDPM formulation operate on the latent representation $z = \bigcup_{i=0}^{b-1} z^i$. Before iterating through each block, z^i, we use
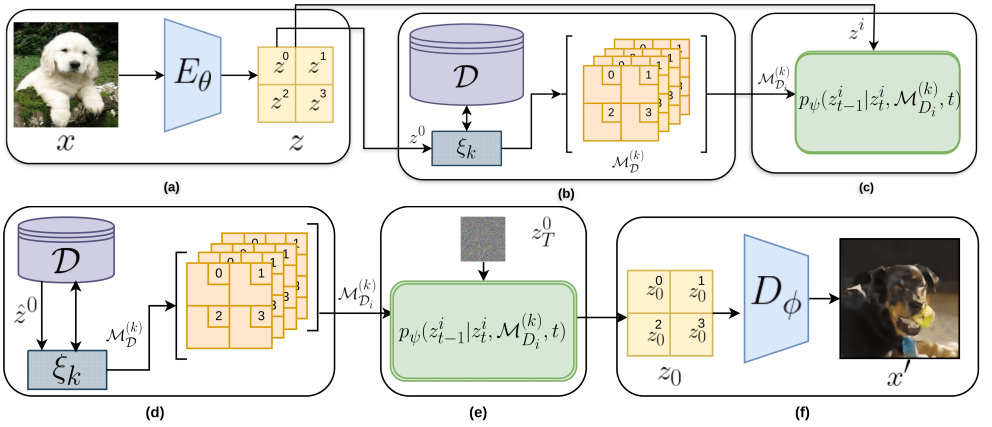
Figure 1: Training and Sampling of RISSOLE: (a) During training, each image x goes through a VQ-GAN encoder E_θ, resulting in the generation of a latent representation z. (b) This latent representation z is then utilized by the retriever $\xi_k(.)$ to fetch its k nearest neighbors $\mathcal{M}_\mathcal{D}^{(k)}$ from a dataset $\mathcal{D}$. (c) Each group of blocks $\mathcal{M}_{\mathcal{D}_i}^{(k)} \in \mathcal{M}_\mathcal{D}^{(k)}$ is used as a conditioning signal for the corresponding z^i, aiding in the training procedure. (d) During the sampling (generation) phase, a pseudo-query $\hat{z}^0$ obtained from the dataset $\mathcal{D}$ is employed by $\xi_k(.)$ to retrieve $\mathcal{M}_\mathcal{D}^{(k)}$. (e) From the retrieved set $\mathcal{M}_\mathcal{D}^{(k)}$, each subset $\mathcal{M}_{\mathcal{D}_i}^{(k)}$ is used as a conditioning signal for the random noise z_t^i at steps $t = T, T-1, \ldots, 3, 2, 1$ to generate the final denoised block z_0^i of $\hat{z}^0$. (f) All denoised representations z_0^i are reshaped to construct z_0, which is then passed through the decoder D_ϕ to yield the reconstructed sample x'.

the first block[2] z^0 to retrieve the k nearest neighbors of z^0 from $\mathcal{D}$, denoted by $\mathcal{M}_\mathcal{D}^{(k)} = \xi_k(z^0, D)$, where $\xi(.)$ is the retrieval function. Note that once $\mathcal{M}_\mathcal{D}^{(k)}$ is obtained, we do not need to perform the retrieval operation again to obtain $\mathcal{M}_{\mathcal{D}_i}^{(k)}$ for subsequent blocks z^i for $i \in 0, 1, 2, \cdots, b-1$, since $\mathcal{M}_\mathcal{D}^{(k)} = \bigcup_{i=0}^{b-1} \mathcal{M}_{\mathcal{D}_i}^{(k)}$. Subsequently, each block z^i is then conditioned on $\mathcal{M}_{\mathcal{D}_i}^{(k)}$ to facilitate the training process through the DDPM U-Net, parameterized by ψ. This approach, characterized by its granularity and reliance on semi-parametric generative modeling [2], can be expressed as

$$p_{\psi, \mathcal{D}, \xi_k}(z) = \prod_{i=1}^{b-1} p_\psi(z^i | \{y | y \in \mathcal{M}_{\mathcal{D}_i}^{(k)}\}) \tag{1}$$

The neighboring information denoted as $\mathcal{M}_{\mathcal{D}_i}^{(k)}$, as retrieved from the database $\mathcal{D}_i$, is typically combined with the input block z^i by simply putting them together along the channel dimension. This combined information is input into a U-Net architecture [15] during the reverse denoising stage within a latent diffusion model. However, we take a different approach where the input and the retrieved neighbors are processed separately through different convolutional layers, resulting in distinct standard embeddings. These embeddings are added to the original input before feeding into the U-Net structure. To maintain consistency and prevent unwanted shifts during training, we apply Layer Normalization [1] to the output. This modification allows us to forgo the costly Cross Attention Mechanism [14] for conditioning the input, resulting in fewer parameters and faster training and sampling times.

---

[2]Another design choice can be to use the entire z to do the retrieval since we have access to the full image and its latent representation. We show some results with this choice in the Appendix.

We train RISSOLE using the standard re-weighted likelihood objective, as described in [10], resulting in the objective similar to the one used in [10, 18]:

$$\min_{\psi} \mathcal{L} = \sum_{i=0}^{b-1} \mathbb{E}_{p(x), z \sim E_\theta(x), \varepsilon \sim \mathcal{N}(0,1), t}[||\varepsilon - \varepsilon_\psi(z_t^i, t, \{y|y \in \mathcal{M}_{\mathcal{D}_i}^{(k)}\})||] \tag{2}$$

In Eq. 2, ε_ψ denotes the UNet-based denoising autoencoder detailed in [15] and applied in [5, 14]. In addition to serving as the objective for a typical DDPM model, Eq. 2 also includes the collection of retrieved neighboring data points as a prerequisite for the noise predictor U-Net: ε_ψ. Furthermore, the loss is summed across all blocks where it is individually computed. The notation $t \sim \text{Uniform}\{1, \ldots, T\}$ indicates the time step, as elaborated in [14, 18]. A pseudocode outlining the training procedure can be found in the Appendix.

4.2   RISSOLE Sampling

Like RDM [2], RISSOLE begins by picking a random *pseudo-query* from the database for generating new samples. However, unlike RDM, which uses a whole image as the query, we use its first block $\hat{z}^0$ from the first block of our database $\mathcal{D}_0$. We then use the query $\hat{z}^0$ to retrieve k nearest neighbors from the database (see Fig. 1 (d)).

Thereafter, we generate a random noise sample from a standard Gaussian. This sample is divided into b fixed-size blocks, each covering a part of the latent space. Next, we perform block-wise sampling, where each block z^i is sampled iteratively conditioned on its k nearest neighbors $\mathcal{M}_{\mathcal{D}_i}^{(k)}$ from the database, following a process similar to the one used during training. Once all blocks have been updated, we reconstruct the complete latent representation by combining the representations from each block. Finally, we decode this combined latent representation using the VQ-GAN Decoder D_ϕ with parameters ϕ to produce a sample x in the original data space. This enables structured sampling from the latent space, capturing relationships within blocks while maintaining coherence across generated samples.

A distinguishing aspect of RISSOLE is that both training and sampling can be parallelized across the b blocks (as the conditioning structure we use does not depend on the blocks on each other), thereby yielding further speed-ups during training and sampling. A pseudocode for sampling from RISSOLE is presented in the Appendix.

5   Experiments

5.1   Dataset and Baselines

We evaluate RISSOLE on two benchmark datasets: CelebA 64×64 [12] and ImageNet100 [4]. CelebA64 contains 202,599 images showing human faces with $64 \times 64 \times 3$ resolution, while ImageNet 100 has 127,878 natural images from 100 categories, each with a resolution of $224 \times 224 \times 3$. We used 80% of the data for training and the remaining 20% for validation, and the same training/validation split was used for both datasets. We compare RISSOLE with RDM [2], a recent SOTA method for compact-sized diffusion models. RDM uses an external database to keep the size of the diffusion model module small. In addition to RDM, we also compare RISSOLE with two other baselines - a variant of RISSOLE that does not use retrieval guidance and another variant that does not use retrieval guidance but uses the different blocks of the image to condition the current block (this is an instance of the recent work on patch-based diffusion models [7]). Please note that these are the most relevant baselines for RISSOLE. Although fast to train, recent methods like patch diffusion [6] are

not directly comparable with RISSOLE since such approaches still require a small set of full-sized images during training. Nevertheless, we compare RISSOLE with Patch Diffusion when it does not use full-sized images during training to ensure fairness in comparison. Thus, their model sizes are not small, unlike RISSOLE, and are thus not included in the comparison here. It is also important to highlight that the roles played by the external database differ in RDM and RISSOLE. In RDM, it helps keep the diffusion model size small, whereas in RISSOLE, the primary role of the external database is to achieve coherence across blocks. In contrast, RISSOLE's block-wise training/generation ability contributes to its parameter efficiency.

5.2 Experimental Setup

In training the VQ-GAN encoder-decoder, we have maintained $f = 8$ (refer to Section 4.1.1) across both datasets. Specifically, for the CelebA dataset, the VQ-GAN's latent dimension is configured to 10. Conversely, due to ImageNet's higher resolution, the latent dimension is adjusted to 32. Consequently, the latent resolution for an image from CelebA will be $10 \times 16 \times 16$, while for ImageNet, it will be $32 \times 28 \times 28$. To retrieve the nearest neighbors from $\mathcal{D}$, we use ScanNN [9] search algorithm in the latent representation generated by a VQ-GAN [8].

In alignment with the requirements of the chosen retrieval mechanism, each latent representation is flattened. For instance, considering the ImageNet100 dataset $\mathcal{X} \subset \mathbb{R}^{224 \times 224 \times 3}$ with $|\mathcal{X}| = n$, and $z \in \mathbb{R}^{28 \times 28 \times 32}$ the dimensionality of D is denoted as $\mathbb{R}^{b \times n \times 6272}$, with each $\mathcal{D}_i \in \mathbb{R}^{n \times 6272}$ where 6272 emerges from the flattening process applied to z. Our models are trained for 200 epochs on an NVIDIA 1080Ti (CelebA64) and an NVIDIA A30 (ImageNet100). The number of nearest neighbors k can influence the performance of retrieval-based approaches. In our experiments, $k = 20$ was found to work best for RDM [2] and $k = 10$ worked best for RISSOLE. In addition, we also conduct an ablation experiment in Sec. 5.4.1 to assess RISSOLE's performance when conditioned with positional information.
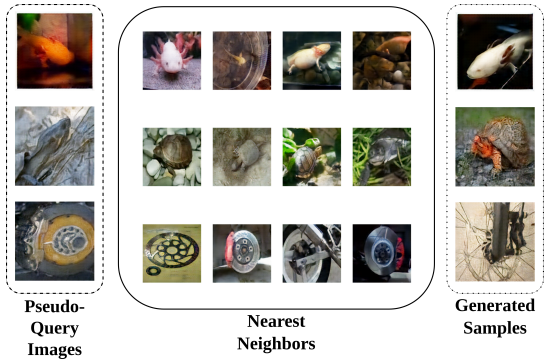

Figure 2: Each of the three rows above show a pseudo-query image $\hat{x}$ (used in generation time) from $\mathcal{D}$, its retrieved neighbors, and the generated sample when these neighbors are conditioned on. Note that the actual training and sampling occur in the latent space. These images, decoded from the latent representations, are for better understanding and visualization.

Pseudo-Query Images **Nearest Neighbors** **Generated Samples**

5.3 Unconditional Image Synthesis

Upon generating samples from our diffusion model, we observed a notable improvement in visual quality compared to the RDM baseline when constrained to roughly the same model size as RISSOLE. Despite its compact model size, the images generated by RISSOLE exhibit finer details, sharper features, and more realistic textures. This qualitative assessment is further supported by quantitative metrics, including the Fréchet Inception Distance (FID).
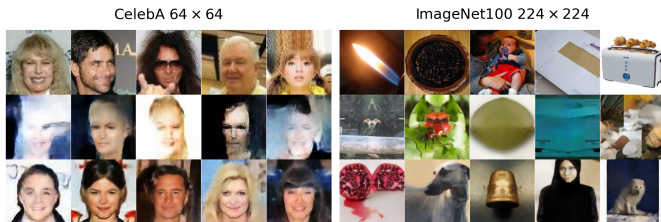
Figure 3: Original Images (top row), and samples generated by the RDM baseline (middle row) and by RISSOLE (bottom row), trained on CelebA and ImageNet 100 datasets.

Table 1 presents the FID scores obtained by RISSOLE and the other baseline models on both the CelebA and ImageNet datasets. RISSOLE achieves significantly lower FID scores, indicating that its generated samples are more likely to be from the original dataset distribution. These results demonstrate that RISSOLE captures the intricate patterns and structures in the data and generates more realistic and coherent images than the other baselines with similar model sizes. To further illustrate the qualitative differences, Fig. 3 compares samples generated by RISSOLE, original images from both datasets and samples from the RDM baseline. As depicted in Fig. 3, RISSOLE produces noticeably better samples, closely resembling the characteristics of the original images. For fairness of comparisons, we aimed to ensure that the model sizes for RDM and RISSOLE are kept similar, as reported in Table 1. To ensure roughly equal model sizes for RDM and RISSOLE, the U-net architecture used by RDM was suitably modified. However, note that because RDM still utilizes the entire image or latent space for training or sampling, it still requires a somewhat higher number of parameters under this constrained setting. Table 1 shows that RISSOLE outperforms RDM on both datasets and Patch Diffusion on CelebA 64×64 [3]. When constrained to small model size, RDM struggles to model the full-sized images with high fidelity and faces difficulty producing coherent and interpretable sample outputs, as shown in the middle row of Fig. 3. In contrast, RISSOLE operates only on small-sized blocks and can, therefore, model these blocks more effectively despite its small model size.

Table 1: Comparison of RISSOLE with RDM[4], Patch Diffusion[22] and other variants of RISSOLE for unconditional image generation on ImageNet100 [4].

Model	CelebA	ImageNet100
RDM	32.36	60.61
Patch Diffusion	14.51	-
RISSOLE - RAG	23.45	29.59
RISSOLE + P	17.01	20.28
RISSOLE + Pos	11.61	14.59
RISSOLE	**9.82**	**12.93**

5.4 Ablation

We also conduct ablation studies to gain insights into the behavior and performance of RISSOLE under different configurations and settings. Specifically, we investigate the impact of varying the number of nearest neighbors retrieved from the database on the model's performance, measured in terms of FID scores. We also vary the number of blocks into which the

---

[3]FID values for Patch Diffusion are from the original paper. No FID results for ImageNet100 were reported.

Figure 4: Qualitative Samples from RISSOLE models where the input is conditioned with (top) and without (bottom) the positional information.

latent space should be divided to observe how the sampling quality changes. Furthermore, we discuss the impact of the retrieval augmented generation to empirically demonstrate how it helps in the coherence of the generated images.

5.4.1 Impact of Positional Information

In Section 4.1.3, we described how RISSOLE learns a specific segment of the latent space, z^i, by conditioning it with $\mathcal{M}_{\mathcal{D}_i}^{(k)}$, while also integrating positional information, represented by i. This section delves into the significance of positional information within RISSOLE, suggesting that such information might already be inherent in $p_\psi(z^i | \mathcal{M}_{\mathcal{D}_i}^{(k)})$. Hence, the model can represented by a modified version of Equation 1:

$$p_{\psi, \mathcal{D}, \xi_k}(z) = \prod_{i=1}^{b-1} p_\psi(z^i | \{y | y \in \mathcal{M}_{\mathcal{D}_i}^{(k)}\}, i) \tag{3}$$

The experiments are conducted using the CelebA dataset with a block count of $b = 4$. Figure 4 demonstrates two sets of samples generated by RISSOLE; the upper row displays samples with included positional information, while the lower row exhibits samples without positional information. Observation of the figure visually confirms that samples generated from RISSOLE without positional conditioning exhibit sharper and more realistic characteristics compared to those with positional encoding. This phenomenon arises from the additional complexity introduced by positional encoding, making it more challenging for the model to estimate the underlying data distribution accurately. While the CelebA samples generated with positional conditioning yield an FID score of 11.93, as reported in Table 1 RISSOLE without positional encoding surpasses this with an FID of **9.82**. This finding substantiates our conjecture that retrieval-augmented generation and block-wise conditioning inherently preserve positional information, consequently enabling RISSOLE to be trained without explicit positional conditioning, thereby enhancing the model's time efficiency.

5.4.2 Using Previous Block as Additional Context

We also explore conditioning on the previously generated block *in addition to* the nearest neighbors from the external database and compare the model's performance with and without this extra conditioning to assess its impact on image synthesis quality. Figure 5 indicates that *also* employing the previous block in conditioning results in a degradation in image quality compared to when it is not

Figure 5: Samples from RISSOLE with (top) and without(bottom) using the previous block as a condition.

used. Specifically, samples generated without utilizing the previous block exhibit finer details, sharper features, and higher fidelity, indicating superior visual quality. FID scores of RISSOLE with the previous block serving as a context (RISSOLE-P) are presented in Table 1. Incorporating the previous block as a condition may impose limitations restricting the model's capacity to comprehend the complete diversity and intricacy of the underlying data distribution. Conversely, training the model without any context from the preceding or other blocks enables the parallelization of the training process, resulting in enhanced speed.

5.4.3 Impact of RAG

We examine the role of the retrieval augmented generation (RAG) mechanism in our diffusion model by comparing the generated samples with and without RAG in Figure 6. When the RAG mechanism is omitted, we observe a degradation in image quality, resulting in incoherent and less visually appealing samples. Table 1 shows the FID scores of a RISSOLE model trained without RAG conditioning, which quantitatively confirms the same.

Figure 6: RISSOLE samples with (bottom) and without (top) using RAG

Incorporating the RAG mechanism is crucial in guiding the block-wise diffusion process and ensuring the coherence and realism of generated images. The retrieved neighbors offer a comprehensive view of the image, aiding the model in generating coherent images block-by-block without needing explicit information about previous blocks. Conditioning the noise on retrieved nearest neighbors through RAG ensures consistency and structural integrity, enhancing the overall fidelity of the synthesized images.

6 Conclusion

In this paper, we presented a block-based retrieval-augmented generation (RAG) as an effective solution for the growing model size of denoising diffusion models. The inclusion of RAG addresses any consistency issues in block-wise image generation. We demonstrate improved spatial and semantic coherence in the generated images by leveraging RAG within a block-wise denoising diffusion model. Our approach, which conditions each image block on a corresponding block from a reference set retrieved from an external database, effectively mitigates the challenges associated with block-wise generation. Our experiments and analysis show that RAG can enhance the quality of block-wise generated images while reducing computational complexity and model size. This approach offers a promising alternative for resource-constrained settings where deploying large generative models may not be feasible. Further research can explore extensions and optimizations of RAG-based block-wise generation methods. Additionally, investigating the applicability of our approach to other domains beyond image generation could uncover new opportunities for leveraging retrieval-augmented techniques for enhanced generative modeling.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016.

[2] Andreas Blattmann, Robin Rombach, Kaan Oktay, Jonas Müller, and Björn Ommer. Semi-parametric neural image synthesis. *arXiv preprint arXiv:2204.11824*, 2022.

[3] Mengyu Chu, You Xie, Jonas Mayer, Laura Leal-Taixé, and Nils Thuerey. Learning temporal coherence via self-supervision for gan-based video generation. *ACM Transactions on Graphics (TOG)*, 39(4):75–1, 2020.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[5] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[6] Zheng Ding, Mengqi Zhang, Jiajun Wu, and Zhuowen Tu. Patched denoising diffusion models for high-resolution image synthesis, 2023.

[7] Zheng Ding, Mengqi Zhang, Jiajun Wu, and Zhuowen Tu. Patched denoising diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024.

[8] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[9] Ruiqi Guo, Philip Sun, Erik Lindgren, Quan Geng, David Simcha, Felix Chern, and Sanjiv Kumar. Accelerating large-scale inference with anisotropic vector quantization. In *International Conference on Machine Learning*, pages 3887–3896. PMLR, 2020.

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[11] Zhifeng Kong and Wei Ping. On fast sampling of diffusion probabilistic models. *arXiv preprint arXiv:2106.00132*, 2021.

[12] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[13] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[14] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 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*, pages 234–241. Springer, 2015.

[16] Robin San-Roman, Eliya Nachmani, and Lior Wolf. Noise estimation for generative diffusion models. *arXiv preprint arXiv:2104.02600*, 2021.

[17] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation, 2023.

[18] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

[19] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[20] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[21] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

[22] Zhendong Wang, Yifan Jiang, Huangjie Zheng, Peihao Wang, Pengcheng He, Zhangyang Wang, Weizhu Chen, and Mingyuan Zhou. Patch diffusion: Faster and more data-efficient training of diffusion models, 2023.

[23] Jingjing Xu, Xuancheng Ren, Yi Zhang, Qi Zeng, Xiaoyan Cai, and Xu Sun. A skeleton-based model for promoting coherence among sentences in narrative story generation. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4306–4315, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1462. URL https://aclanthology.org/D18-1462.

[24] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Comput. Surv.*, 56(4), nov 2023. ISSN 0360-0300. doi: 10.1145/3626235. URL https://doi.org/10.1145/3626235.

[25] Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, and Bin Cui. Retrieval-augmented generation for ai-generated content: A survey. *arXiv preprint arXiv:2402.19473*, 2024.