Learning 3D Volume Cloud from Single Image

Yuhang Cheng
Southwest University
Chongqing, China
chengyuhang1999@email.swu.edu.cn

Yu Zhang*
Southwest University
Chongqing, China
zhangyuhetiannan@gmail.com

Xiaogang Wang*
Southwest University
Chongqing, China
wangxiaogang@swu.edu.cn

Abstract

Three-dimensional (3D) cloud modeling plays a pivotal role in advancing atmospheric models and enhancing natural phenomena visualization systems. Nevertheless, the high-quality reconstruction of clouds remains a significant challenge, primarily due to their inherent heterogeneous nature as volumetric media. Image-based modeling approaches offer a promising solution to this challenge. This paper presents a novel two-stage neural network architecture for 3D cloud reconstruction from single image. The first stage introduces an innovative view synthesis network built upon Stable Diffusion, incorporating two specialized modules: a cloud mapper and a viewpoint mapper, which collaboratively generate novel perspective views from a single input image. The second stage implements a physics-based differentiable rendering framework to construct a 3D cloud reconstruction network, leveraging the synthesized multi-view images to optimize a volumetric density grid representation. To enhance the reconstruction fidelity, we integrate real-world cloud density distribution statistics and implement a post-processing refinement using Perlin-Worley noise combined with Fractal Brownian Motion (FBM) for erosion effects. Additionally, to mitigate the inherent limitations of geometric information extraction from single-view images, we developed a comprehensive cloud simulation dataset for pre-training the viewpoint mapper module. This dataset encompasses multi-view cloud images with corresponding camera extrinsic parameters, capturing a diverse range of cloud formations. Extensive quantitative evaluations and qualitative assessments demonstrate the efficacy and potential of our proposed two-stage network in achieving accurate 3D cloud reconstruction from single-view images.

CCS Concepts

• **Computing methodologies** → **Shape modeling**.

Keywords

Cloud modeling, single-view reconstruction, stable diffusion, density grid

*Corresponding authors.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ICMR '25, Chicago, IL, USA
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-1877-9/2025/06
https://doi.org/10.1145/3731715.3733371

ACM Reference Format:
Yuhang Cheng, Yu Zhang, and Xiaogang Wang. 2025. Learning 3D Volume Cloud from Single Image. In *Proceedings of the 2025 International Conference on Multimedia Retrieval (ICMR '25), June 30-July 3, 2025, Chicago, IL, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3731715.3733371

1 Introduction

Clouds play a crucial role in Earth's atmospheric processes and climate system. Remote sensing techniques provide essential data for cloud observation, enabling the acquisition of various cloud properties critical for 3D reconstruction and modeling. Accurate 3D representations of clouds are essential for enhancing atmospheric models and improving the retrieval of cloud properties from satellite observations. However, modeling clouds with specific shapes remains challenging due to their complex and heterogeneous nature. Consequently, research on cloud modeling is still in its developmental stage, with only a few published studies dedicated to this field. These methods can be broadly categorized into three approaches: process-based, physics-based, and image-based.

The process-based method [1–5] refers to cloud modeling based on mathematical models, which induce cloud shapes' characteristics to synthesize relatively realistic cloud scenes in a comparatively short time. However, this approach requires the determination of many parameters through trial and error, which is inconvenient. To address this issue, the physics-based method [6–9] models clouds by simulating the cloud generation process. Nevertheless, this typically demands significant computational time. Furthermore, both aforementioned methods have limitations in modeling specific clouds.

In response to these challenges, researchers have proposed image-based methods [10–13] for cloud modeling, which align with our objective. Unlike the first two categories of approaches, image-based methods aim to utilize geometric information from the input cloud image to model clouds. It is important to note that all image-based methods use the image as a guide to synthesize clouds that appear similar to those in the image, rather than reproducing exactly the same clouds. This limitation arises from the inherent ambiguity in describing three-dimensional objects using single images.

Originally, Dobashi et al. utilized various algorithms to model different types of clouds from a single image [10]. Their 3D modeling process is based on the assumption that a cloud's interior is denser than its boundary and exhibits symmetry with respect to the image plane. Nevertheless, this approach fails to capture the intricacies of the cloud's surface, as it focuses solely on the overall shape. To overcome this limitation, Yuan et al. generated a 3D shape by inverse solving of a single scattering model [6] to model cumulus clouds from a single image [11]. Their approach also assumes cloud shape symmetry and sets simple lighting parameters, such as the extinction coefficient and albedo, as constants. However, many clouds do not conform to these assumptions. Moreover, the side

surface of the cumulus cloud remains unnatural due to the stitching of only the front and back surfaces. To enhance the reconstruction of surface details, Zhang et al. proposed a Shape from Shading (SFS) algorithm [14, 15] based on unknown illumination, aiming to reconstruct the cloud surface from a given image [12]. They incorporated boundary constraints and relative height constraints as prior information to improve the reconstruction process. However, meteorological knowledge indicates that clouds are non-uniform participating media, making direct surface reconstruction unreasonable and resulting in reconstructed outcomes that fail to meet the requirements of downstream tasks. In recent years, Zhang et al. have proposed a deep learning-based approach to reconstruct 3D cloud objects from a single image [13]. They introduced a Variational Autoencoder (VAE) [16] and Generative Adversarial Network (GAN) [17] to learn the latent space of 3D cumulus shapes. Although this approach somewhat abandons simple priors such as symmetry used in traditional methods, it unreasonably defines cloud objects as surface meshes.

Unlike previous methods, this paper approaches the problem from the perspective of cloud physical properties and proposes a two-stage network based on data-driven priors, aiming to reconstruct three-dimensional cloud objects from a single image. In summary, the main contributions of this work are as follows:

- We propose a novel viewpoint synthesis network designed to generate new perspective images of a cloud from a single cloud image. This network comprises a cloud object mapper and a viewpoint mapper. The former is responsible for learning the unique surface features of the input cloud object, while the latter constrains the generation of new perspective images based on geometric priors obtained from a self-constructed volumetric cloud dataset.
- We introduce a cloud three-dimensional object reconstruction network aimed at generating a 3D cloud volume grid from multi-view images obtained from the novel viewpoint synthesis network. Considering that clouds are highly albedo, inhomogeneous participating media, we utilize a physics-based differentiable renderer for volume grid optimization. The optimized cloud object can be used for downstream tasks such as relighting.
- We construct a multi-view cloud synthesis dataset. To support the implementation of our data-driven prior-based approach and address the absence of publicly available cloud datasets, we have created a cloud synthesis dataset. This dataset contains multiple clouds, each comprising multi-view images and corresponding camera extrinsics.

2 Related Work

This section provides a concise overview of existing cloud modeling methodologies, which can be broadly categorized into procedural-based, physically-based, and image-based approaches. Among these, procedural modeling has been the most widely adopted technique for cloud synthesis. Early contributions by Voss et al. utilized fractals for cloud modeling [18], while Perlin et al. introduced the use of Perlin noise for generating realistic cloud textures [19]. Sakas et al. further advanced the field by employing spectral synthesis for cloud modeling [20]. Building on these foundations, Schpok et al. developed a real-time system for procedural cloud modeling [21]. More recently, Elhaddad et al. proposed an innovative cloud simulation method that leverages an approximate Lennard-Jones potential to simplify and simulate cloud dynamics [22]. Schneider et al., Elhaddad et al., and Hillaire et al. combined Worley noise and Perlin noise with ray marching techniques to achieve realistic rendering of cumulus clouds under dynamic lighting conditions in real-time [3–5]. Despite the impressive visual results achieved by these methods, they often require extensive parameter tuning through trial and error, which limits their practicality.

To address these limitations, physically-based methods have been proposed to simulate the cloud formation process by solving the Navier-Stokes equations and thermodynamic principles [6–8]. While these methods are capable of generating highly realistic cloud formations, they are computationally intensive and often impractical for real-time applications.

In contrast to procedural and physically-based approaches, image-based cloud modeling methods focus on reconstructing specific cloud formations from images, which aligns closely with our research objectives. Early work by Dobashi et al. introduced various modeling techniques for different cloud types [10]. However, their approach relies on simplified assumptions about cloud thickness, which, while capable of producing visually plausible results, fails to capture fine-scale geometric details such as concave and convex surface features. Yuan et al. later proposed a computational model for reconstructing cumulus cloud shapes from single images [11]. This method, however, assumes cloud volume symmetry, which contradicts empirical observations of natural clouds that often exhibit asymmetric structures. Zhang et al. advanced the field by introducing a Shape from Shading (SFS) algorithm [14, 15] to reconstruct cloud surfaces from single images [12]. Their approach incorporates boundary constraints and relative height constraints as prior information to improve reconstruction accuracy. Nevertheless, given that clouds are non-uniform participating media, direct surface reconstruction remains inherently limited and often yields results unsuitable for downstream applications.

With the advent of deep learning, Zhang et al. proposed a three-dimensional autoencoder network to learn the latent space of 3D cumulus cloud shapes, aiming to address the problem of 3D cloud reconstruction from single images [13]. While this approach eliminates the need for traditional geometric priors, it represents clouds using surface meshes, which are inherently unsuitable for capturing the volumetric nature of clouds. Consequently, the reconstructed cloud geometries exhibit low-frequency surface details and are often inadequate for downstream tasks.

3 Method

Given a single input image, we divide the 3D volumetric cloud modeling process into two stages: novel view synthesis (Section 3.3) and 3D reconstruction (Section 3.4). Before delving into the details of our method, we first provide a brief background on text-to-image Stable Diffusion and textual inversion (Section 3.1). Subsequently, we describe the construction of our volumetric cloud synthesis dataset, which is essential for training our model (Section 3.2).

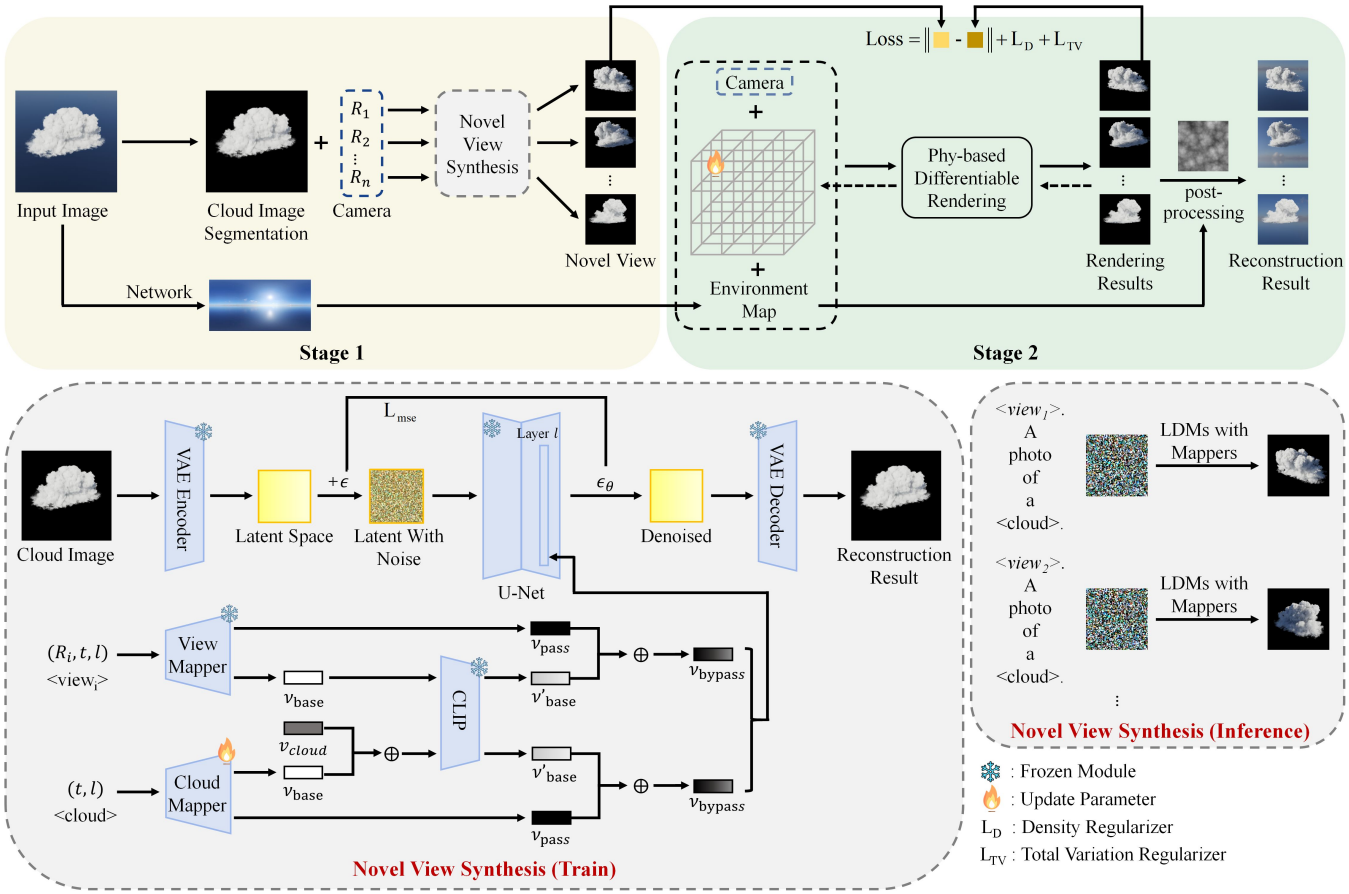

Figure 1: The two-stage pipeline of our single-view 3D volumetric cloud reconstruction method. Stage 1 generates multi-view cloud images from a single input image, while Stage 2 reconstructs the 3D density grid using these multi-view images and applies post-processing to enhance surface details.

3.1 Preliminaries

3.1.1 Stable Diffusion (SD) [23]. Stable Diffusion (SD) is a state-of-the-art generative model for image synthesis, trained on large-scale datasets to learn a denoising autoencoder ϵ_θ. This autoencoder predicts a denoised version of a noisy latent z_t at each timestep t during the diffusion process. For text-to-image generation, the input image x is first encoded into a lower-dimensional latent space $z = \varepsilon(x)$ using a variational autoencoder (VAE) [16]. After the denoising process, the latent representation is decoded back into pixel space via the VAE decoder. The training objective for SD is formulated as:

$$L = \mathbb{E}_{z\sim\varepsilon(x),y,\epsilon\sim\mathcal{N}(0,1),t} \|\epsilon - \epsilon_\theta(z_t, t, c(y))\|_2^2, \quad (1)$$

where ϵ_θ is a conditional UNet [24], z_t is the noisy latent obtained by adding noise to z at timestep t, and $c(y)$ is the conditioning vector derived from text prompts y using a pretrained CLIP model [25]. During the denoising process, z_t is processed through the UNet, while $c(y)$ is integrated into the network via cross-attention layers.

3.1.2 Neural Representation for Textual Inversion [26]. While Stable Diffusion (SD) excels at capturing basic semantic information in text-to-image tasks, it often struggles to accurately represent the appearance of specific concepts. This limitation leads to significant visual discrepancies when generating images of specific themes. Retraining the model for each new concept is computationally expensive, and fine-tuning with a few example images often results in catastrophic forgetting. To address these challenges, Textual Inversion (TI) and its variants have been developed.

In TI, Gal et al. [27] learn an embedding vector v_* that represents a new concept from a small dataset of 3-5 user-provided images. The optimization process minimizes the loss in Equation (1) while keeping the diffusion model weights frozen:

$$v_* = \min_v \mathbb{E}_{z,y,\epsilon,t} \|\epsilon - \epsilon_\theta(z_t, t, c(y, v))\|_2^2, \quad (2)$$

where $c(y, v)$ is the conditioning vector incorporating the optimized embedding v.

Building on TI, Alaluf et al. introduced Neural Textual Inversion (NeTI), which enhances the quality and controllability of generated compositions. Instead of using a single embedding vector, NeTI employs a neural mapper M to learn embeddings for each denoising

timestep t and U-Net layer l. The objective is defined as:

$$\min_{M} \mathbb{E}_{z,y,\epsilon,t} \|\epsilon - \epsilon_\theta(z_t, t, c(y, M(t,l)))\|_2^2. \quad (3)$$

Recently, Burgess et al. proposed Viewpoint Neural Textual Inversion (ViewNeTI) [28] for novel view synthesis from sparse viewpoints. ViewNeTI controls the viewpoint of objects in images generated by text-to-image SD and is based on NeTI. It consists of two neural networks: a pre-trained viewpoint mapper M_v and an object mapper M_o. The viewpoint mapper, trained on multi-view images of multiple objects, controls the viewpoint in the generated images, while the object mapper describes the concept of the input image. Inspired by ViewNeTI, we construct our novel view synthesis network based on pre-trained SD, with details provided in Section 3.3.

3.2 3D Cloud Dataset Creation

Reconstructing the 3D structure of a cloud from a single image is an ill-posed problem, primarily due to the lack of geometric information from unseen viewpoints. While humans can intuitively infer the 3D shape of a cloud from a single image [29], such mental reconstructions often deviate from the actual structure. To address this, we aim to enhance the quality of 3D reconstruction from single-view images by incorporating geometric priors of clouds into our model. This requires a dataset of multi-view cloud images for training. However, to the best of our knowledge, no publicly available dataset meets this requirement. Therefore, constructing a high-quality, photorealistic cloud dataset for deep learning applications is of significant practical importance.

We utilize the CloudScapes library [30], a photorealistic 3D volumetric cloud dataset in VDB format [31], to create our multi-view cloud dataset. This dataset includes cumulus, cumulonimbus, and stratocumulus clouds. The image generation process is implemented using the Cycles rendering engine in Blender [32] and consists of three key steps: lighting setup, cloud rendering configuration, and camera position selection.

- *Lighting setup:* We download high-quality HDR images from Poly Haven [33] to simulate realistic lighting conditions.
- *Cloud rendering configuration:* The bounding box of the VDB cloud data is normalized, and the cloud's center is positioned at $(0, 0, 200)$ in the Blender world coordinate system.
- *Camera position selection:* Camera positions are sampled on the surface of a sphere with a radius of 2.5, centered at $(0, 0, 200)$. The azimuthal angle ranges from -120° to 120°, and the polar angle ranges from 65° to 115°, with 50 positions uniformly sampled.

Using this pipeline, we generate a dataset of 37 distinct cloud scenes, totaling 1850 rendered images. Each image has a resolution of 512×512 pixels in RGBA format, and the corresponding camera extrinsic parameters are saved for each rendering. Examples from the dataset are shown in Figure 2.

3.3 Novel View Synthesis

Given a single cloud image $x \in \mathbb{R}^{H \times W \times 3}$, segmented from the input image using CarveKit [34] to retain only the foreground object, our goal is to synthesize images of the cloud from different viewpoints. Let R_i denote the extrinsic parameters of a new viewpoint. We aim to learn a model f that generates the cloud image under R_i:

$$\hat{x}_{R_i} = f(x, R_i), \quad (4)$$

where $\hat{x}_{R_i}$ is the synthesized novel view image. For simulated cloud images input to the network, the camera parameters are known. This approach is also applicable to real cloud data, as knowing the camera parameters is essential for 3D reconstruction and downstream tasks. These parameters are either pre-known or manually specified by the user [10, 11]. For cases where the camera parameters are unknown, Dobashi et al. proposed a simple yet effective assumption: the camera's upward direction is perpendicular to the z-axis, and its line of sight is parallel to the x-axis. Based on this assumption, users only need to specify an additional viewing angle to calculate the camera's extrinsic parameters. Given the input single cloud image, users typically have a clear understanding of the approximate viewing angle.

Generating novel viewpoints from a single image is highly challenging due to the limited information available. Recent advancements have shown that Stable Diffusion (SD), trained on internet-scale datasets, is capable of 3D reasoning [29, 35, 36]. Leveraging this capability, we employ SD to synthesize novel views from a single input cloud image.

Our approach utilizes two small neural networks: a cloud mapper (M_{cloud}) and a view mapper (M_{view}). These networks generate token embeddings for the input cloud object and its viewpoint, respectively, which condition the diffusion generation process to control the viewpoints of the input cloud. The cloud mapper, M_{cloud}, takes the diffusion timestep t and UNet conditioning layer l as input. This input is transformed into a high-dimensional vector using positional encoding [37], followed by a 2-layer MLP with 64 dimensions, LayerNorm [38], and LeakyReLU [39]. The output is passed through a final fully-connected layer to produce two 768-dimensional vectors, v_{base} and v_{pass}, representing the concept at the current timestep and layer.

The vector v_{base} is fed into the text encoder to obtain the conditioning vector v'_{base}. The vector v_{pass} is added as a residual to v'_{base} to enhance the flexibility of the mappers' control. The final vector injected into the U-Net cross-attention layer is defined as:

$$v_{bypass} = v'_{base} + \alpha \left| v'_{base} \right| \frac{v_{pass}}{|v_{pass}|}, \quad (5)$$

where α is a scale factor controlling the perturbation magnitude, set to 5 as in ViewNeTI. The view mapper, M_{view}, follows the same architecture as M_{cloud} but additionally takes camera parameters as input for positional encoding.

While SD is trained on diverse objects from various viewpoints, it does not explicitly encode viewpoint correspondences [29]. To address this, we pre-train a view mapper M_{view} using multi-view cloud data. This mapper is shared across all cloud scenes and generalizes to new cloud objects. During pre-training, images of multiple cloud objects under various camera extrinsics are used. The pipeline is similar to the novel view synthesis part in Figure 1, with the view mapper undergoing parameter updates.

To further enhance the quality of novel view synthesis, we implement two additional optimizations:

(1) *Data Augmentations:* To mitigate overfitting, we apply data augmentation to the input cloud image. With a probability of 0.2,



Figure 2: Examples from our multi-view volumetric cloud dataset. Each column shows different cloud instances rendered from the same viewing angle, while each row displays a single cloud instance rendered from multiple viewing angles. This dataset is crucial for injecting cloud geometry prior information into our model. All scenes are rendered using the same environment map to ensure consistency in lighting conditions.

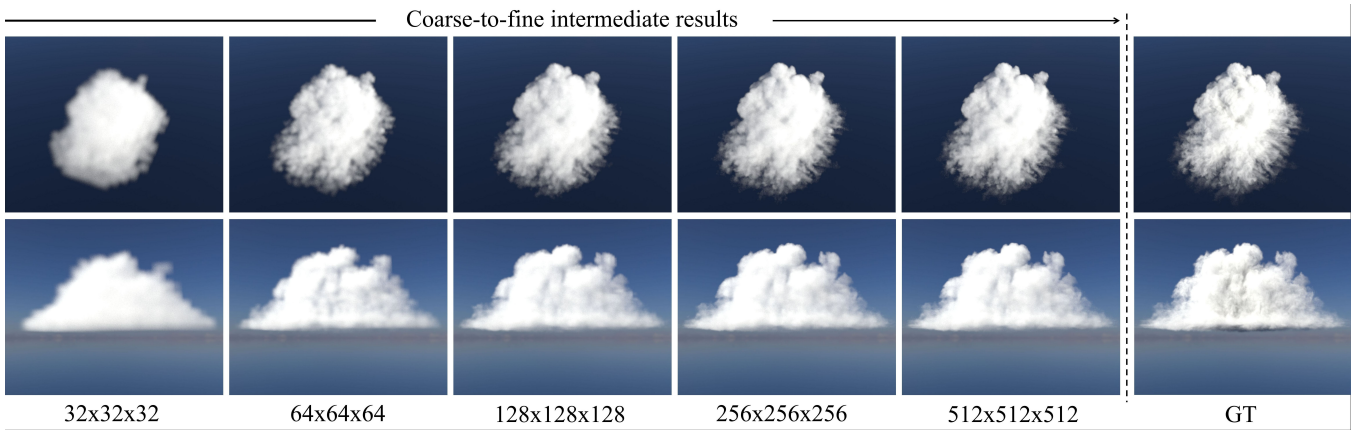

Figure 3: Visualization of intermediate results in our coarse-to-fine 3D volumetric cloud reconstruction method. The figure illustrates the progressive refinement from the initial coarse estimation to the final detailed reconstruction.

a Gaussian filter (kernel size 5) is applied, followed by random cropping. The cropped image is resized to $(512, 384)$ and fed into the network. The random cropping scale is restricted to $(0.9, 1.1)$ to ensure consistency among multi-view images during 3D reconstruction. Figure 10 and Table 1 demonstrate the effectiveness of this approach.

(2) *Cloud Prior Knowledge:* Diffusion models trained on internet-scale datasets inherently capture the data distribution of clouds. In a simple experiment, we generate cloud images using SD with the prompt *"A photo of a cloud"* (see Figure 4). Leveraging this prior knowledge, we rescale the output of the cloud mapper using the norm of the token embedding for *"cloud"* (v_{cloud}). This improves the

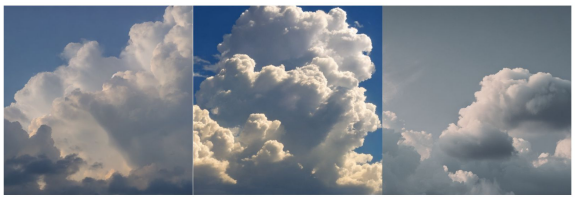

Figure 4: Cloud images generated using Stable Diffusion with the prompt: *"A photo of a cloud"*.



Figure 5: Visualization of the impact of different albedo values on the same cloud density distribution.

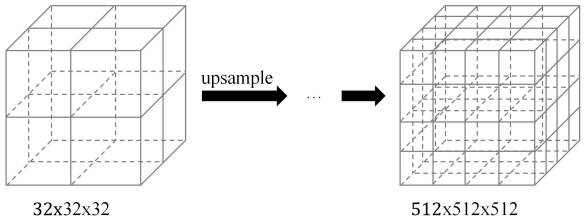

Figure 6: Illustration of the multi-scale grids used in our coarse-to-fine 3D volumetric cloud reconstruction method.

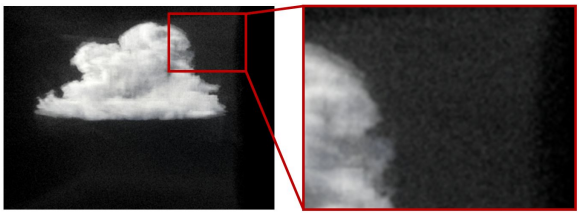

Figure 7: Reconstructed results using only L_{mse}. Discrete density regions are highlighted for clarity.

quality of novel view images and accelerates network convergence, as v_{cloud} carries the optimized data distribution of clouds.

3.4 3D Reconstruction

In this section, we aim to reconstruct the 3D structure of clouds from the multi-view images generated in Stage 1 (Section 3.3). To achieve high-quality reconstruction, we incorporate the following physical properties of clouds into our approach:

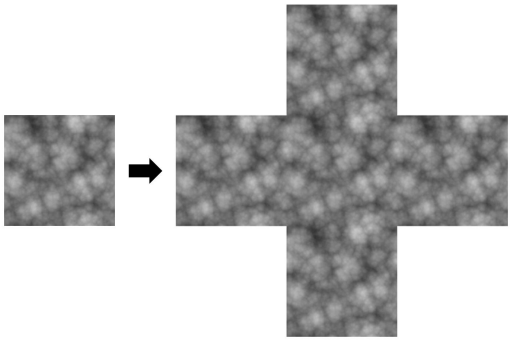

Figure 8: The cross-sectional image of a 3D seamless noise texture.

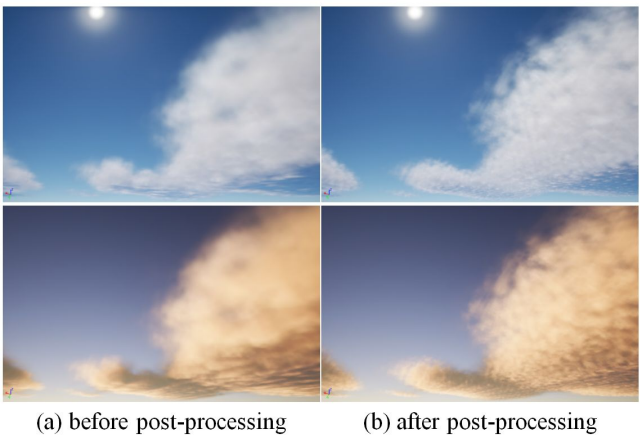

Figure 9: Visualization of cloud density erosion using Perlin-Worley noise in Unreal Engine 4 (UE4).

1) *Clouds are inhomogeneous participating media:* Unlike surface-based objects, clouds cannot be accurately represented using surface meshes. Instead, we model them as volumetric entities using regular voxel grids, where each vertex stores a density value. This representation captures the complex internal structure of clouds more effectively.

2) *Clouds are highly albedo materials:* Albedo quantifies the ratio of scattered to absorbed light within a medium, ranging from $[0, 1]$. A value close to 0 indicates high absorption (e.g., dark smoke), while a value close to 1 indicates high scattering (e.g., clouds or air) [4]. In our experiments, we set the albedo to 1 to reflect the highly scattering nature of clouds.

3) *Clouds exhibit multiple scattering:* To accurately simulate the complex light interactions within clouds, we employ the path-reuse backpropagation (PRB) algorithm [40], implemented in the Mitsuba rendering framework [41]. This algorithm provides an unbiased backward propagation method for rendering, requiring constant memory and offering computational efficiency that scales linearly with the number of scattering events, akin to path tracing.

To summarize, leveraging the physical properties of clouds and to expedite training, we design a coarse-to-fine network for 3D

volumetric cloud reconstruction. Specifically, we employ a multi-scale voxel grid representation for the cloud object, beginning with a resolution of $32\times32\times32$. Subsequent stages involve upsampling the voxel grid from the previous stage (see Figure 6). Multi-view cloud images are used as constraints to optimize the density values of each voxel using the PRB algorithm. The optimized grid, representing the reconstructed cloud volume, is suitable for downstream tasks such as relighting (see Figure 11).

Initially, we optimized the network using only the mean squared error (MSE) loss (L_{mse}). However, this approach led to discrete density regions in the reconstructed output (see Figure 7) and minor inconsistencies in viewpoint alignment from Stage 1. To mitigate these issues, we introduced two density prior regularization terms, L_D and L_{TV}, which significantly improved reconstruction quality by reducing artifacts.

The density regularization term L_D is defined as:

$$L_D = \frac{1}{N}\sum_{x,y,z}\sigma, \tag{6}$$

where σ represents the density value at each voxel, constrained to the range $[0.0, 1.0]$, and $N = x \times y \times z$ is the total number of voxels.

The total variation regularization term L_{TV} is defined as:

$$\begin{aligned} L_{TV} = \sum_{x,y,z} \big(&|\sigma_{x+1,y,z} - \sigma_{x,y,z}| + |\sigma_{x,y+1,z} - \sigma_{x,y,z}| \\ &+ |\sigma_{x,y,z+1} - \sigma_{x,y,z}|\big). \end{aligned} \tag{7}$$

Here, L_D encourages sparsity in the density values, reducing discrete artifacts, while L_{TV} enforces smoothness by penalizing differences between neighboring voxels, serving as a smoothness prior for cloud objects.

Finally, the grid data is optimized from multiple viewpoints using the following combined loss function:

$$L = L_{mse} + L_D + L_{TV}. \tag{8}$$

Post-processing. Since Stage 1 generates multi-view images from a single input image, its primary limitation is the production of rough geometric contours with insufficient fine details. To address this, we apply post-processing to the reconstructed 3D volumetric cloud density grid. Inspired by Schneider et al. [3], we utilize Perlin-Worley noise [42, 43] generated with fractional Brownian motion (FBM) to erode the cloud density. This approach enhances surface details by leveraging visual priors derived from the cloud surface. Figure 8 illustrates a 2D cross-section of the 3D seamless noise texture used in this process.

To validate the effectiveness of our method, we procedurally modeled volumetric clouds in Unreal Engine 4 (UE4) and applied the constructed noise to erode cloud density. Figure 9 demonstrates that the noise effectively simulates realistic cloud surface density variations, confirming its utility in enhancing visual fidelity.

4 Experiments

We evaluate the performance of our model in two key tasks: novel view synthesis from a single viewpoint and 3D reconstruction. It is important to note that research on single-view cloud reconstruction is still in its early stages. Existing methods, such as those in [10–13], focus on different reconstruction objectives and are not directly comparable to our approach. Furthermore, there are no publicly available datasets tailored for this task. As a result, we compare our method with the baseline ViewNeTI [28] on a custom-built dataset.

Table 1: Quantitative comparison of single-image novel view synthesis metrics.[1]

Method	LPIPS	SSIM	PSNR
ViewNeTI	0.583	0.288	8.309
Our	**0.249**	**0.704**	**13.89**

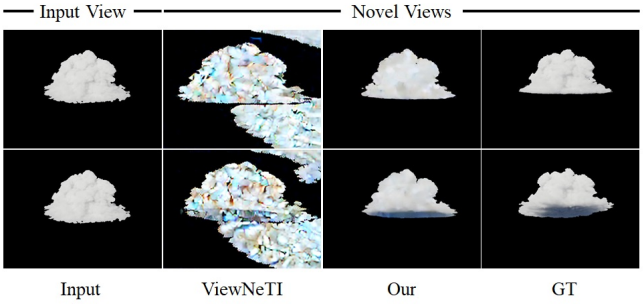

Figure 10: Qualitative comparison of single-view novel view synthesis.

4.1 Datasets and Metrics

Datasets. We evaluate our method on a custom volumetric cloud dataset comprising 37 photo-realistic volumetric clouds, each rendered from 50 viewpoints with known camera extrinsics. From these, 34 cloud instances are used to pre-train the view mapper M_{view}, while the remaining 3 are reserved for evaluation and downstream tasks.

Metrics. For novel view synthesis, we employ three widely-used metrics to evaluate image similarity: PSNR, SSIM [44], and LPIPS [45]. These metrics provide a comprehensive assessment of reconstruction quality. For 3D reconstruction, we present qualitative visual results due to the lack of ground truth volumetric cloud data.

4.2 Novel View Synthesis Results

Table 1 presents the quantitative comparison between our method and the baseline ViewNeTI. Figure 10 provides a qualitative comparison, demonstrating the superiority of our approach. We observe that the baseline model tends to generate extraneous artifacts outside the intended cloud structure. In contrast, our method produces results that are geometrically more consistent with the reference images and better preserve lighting information.

However, our generated results exhibit fewer geometric details compared to the reference images. This is primarily due to two factors: (1) Gaussian smoothing applied during data augmentation to prevent overfitting, and (2) the inherent limitation of extracting detailed density information from a single-view image. These limitations are addressed in Stage 2 through post-processing techniques that enhance surface details.

[1]For LPIPS, lower is better (↓). For SSIM and PSNR, higher is better (↑).

Table 2: Ablation studies on novel view synthesis.[2]

Components		Metrics		
Data Augmentations	Cloud Prior	LPIPS	SSIM	PSNR
		0.583	0.288	8.309
✓		0.312	0.635	12.676
✓	✓	**0.249**	**0.704**	**13.89**

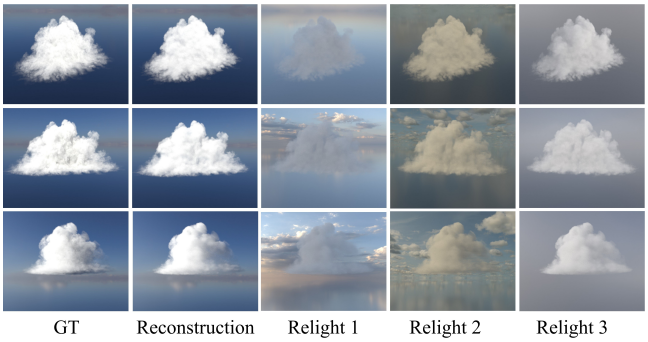

Figure 11: Results of our 3D cloud reconstruction and relighting method applied to the multi-view cloud dataset. The last three columns show relighting results under different environment maps.

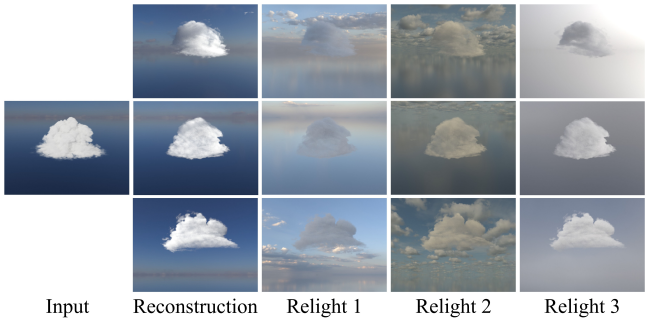

Figure 12: 3D reconstruction and relighting results on multi-view images synthesized in Stage 1. The last three columns demonstrate relighting under varying environment maps.

4.3 3D Reconstruction Results

Figure 11 showcases the reconstruction results of our 3D reconstruction network on the multi-view cloud dataset, while Figure 12 presents the results on the multi-view images generated in Stage 1. Qualitatively, our method reconstructs high-fidelity 3D volumetric clouds that align well with the ground truth. Furthermore, the reconstructed volumetric data is highly suitable for downstream tasks, such as relighting. We demonstrate this by re-rendering the reconstructed volumetric data using three sets of new environment maps (see the last three columns in Figure 11 and Figure 12).

[2]For LPIPS, lower is better (↓). For SSIM and PSNR, higher is better (↑).

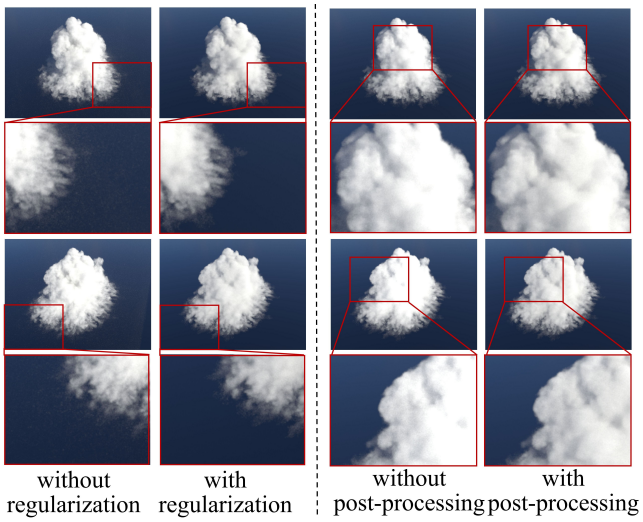

Figure 13: Ablation studies on our 3D cloud reconstruction method, demonstrating the impact of each module on reconstruction quality.

4.4 Ablation Studies

To evaluate the contribution of each module in our 3D reconstruction pipeline, we conduct ablation studies by incrementally adding components. Starting with only the MSE loss (L_{mse}) between the reconstructed and reference images, we sequentially incorporate the density regularization term (L_D), the total variation regularization term (L_{TV}), and the post-processing module. As shown in Figure 13, the addition of each module progressively reduces discrete density artifacts and enhances surface details. The post-processing module, in particular, significantly improves the geometric fidelity of the reconstructed clouds.

5 Conclusion

In this paper, we present a novel two-stage network for reconstructing a 3D volumetric cloud density grid from a single-view input image. Our approach leverages data-driven priors and incorporates the physical properties of clouds as highly albedo, inhomogeneous participating media. This ensures that the reconstruction results are not only accurate but also directly applicable to downstream tasks such as relighting and simulation. To enhance the model's ability to capture geometric priors, we constructed a comprehensive multi-view volumetric cloud dataset. Extensive experiments on this dataset demonstrate the effectiveness of our method, highlighting its potential to address the challenging problem of single-view volumetric cloud reconstruction.

6 Acknowledgment

This work was supported in part by the Chongqing Natural Science Foundation (CSTB2024NSCQ-MSX1026), in part by the Fundamental Research Funds for the Central Universities (No. SWU120076).

References

[1] G. Y. Gardner, "Visual simulation of clouds," in *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, 1985, pp. 297–304.

[2] P. Goswami and F. Neyret, "Real-time landscape-size convective clouds simulation," in *Proceedings of the 19th Symposium on Interactive 3D Graphics and Games*, 2015, pp. 135–135.

[3] A. Schneider and N. Vos, "The real-time volumetric cloudscapes of horizon: Zero dawn," *Advances in Real-time Rendering, SIGGRAPH*, 2015.

[4] S. Hillaire, "Physically based sky, atmosphere and cloud rendering in frostbite," in *ACM SIGGRAPH*, 2016, pp. 1–62.

[5] A. Schneider, "Nubis: authoring real-time volumetric cloudscapes with the decima engine," *SIGGRAPH Advances in Real-Time Rendering in Games Course. ACM*, pp. 619–620, 2017.

[6] J. T. Kajiya and B. P. Von Herzen, "Ray tracing volume densities," *ACM SIGGRAPH computer graphics*, vol. 18, no. 3, pp. 165–174, 1984.

[7] R. Miyazaki, S. Yoshida, Y. Dobashi, and T. Nishita, "A method for modeling clouds based on atmospheric fluid dynamics," in *Proceedings Ninth Pacific Conference on Computer Graphics and Applications. Pacific Graphics 2001.* IEEE, 2001, pp. 363–372.

[8] Y. Dobashi, K. Kusumoto, T. Nishita, and T. Yamamoto, "Feedback control of cumuliform cloud formation based on computational fluid dynamics," *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3, pp. 1–8, 2008.

[9] C. W. Ferreira Barbosa, Y. Dobashi, and T. Yamamoto, "Adaptive cloud simulation using position based fluids," *Computer Animation and Virtual Worlds*, vol. 26, no. 3-4, pp. 367–375, 2015.

[10] Y. Dobashi, Y. Shinzo, and T. Yamamoto, "Modeling of clouds from a single photograph," in *Computer Graphics Forum*, vol. 29, no. 7. Wiley Online Library, 2010, pp. 2083–2090.

[11] C. Yuan, X. Liang, S. Hao, Y. Qi, and Q. Zhao, "Modelling cumulus cloud shape from a single image," in *Computer Graphics Forum*, vol. 33, no. 6. Wiley Online Library, 2014, pp. 288–297.

[12] Y. Zhang, Z. Zhang, J. Hou, and X. Liang, "Estimating cumulus cloud shape from a single image," in *Advances in Image and Graphics Technologies: 12th Chinese conference, IGTA 2017, Beijing, China, June 30–July 1, 2017, Revised Selected Papers 12.* Springer, 2018, pp. 147–156.

[13] Z. Zhang, Y. Cen, F. Zhang, and X. Liang, "Cumulus cloud modeling from images based on vae-gan," *Virtual Reality & Intelligent Hardware*, vol. 3, no. 2, pp. 171–181, 2021.

[14] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah, "Shape-from-shading: a survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 21, no. 8, pp. 690–706, 1999.

[15] J.-D. Durou, M. Falcone, and M. Sagona, "Numerical methods for shape-from-shading: A new survey with benchmarks," *Computer Vision and Image Understanding*, vol. 109, no. 1, pp. 22–43, 2008.

[16] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[17] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[18] R. Voss, "Fourier synthesis of gaussian fractals: 1/f noises, landscapes, and flakes," *Tutorial on State of the Art Image Synthesis, SIGGRAPH'83*, 1983.

[19] K. Perlin, "An image synthesizer," *ACM Siggraph Computer Graphics*, vol. 19, no. 3, pp. 287–296, 1985.

[20] G. Sakas, "Modeling and animating turbulent gaseous phenomena using spectral synthesis," *The Visual Computer*, vol. 9, pp. 200–212, 1993.

[21] J. Schpok, J. Simons, D. S. Ebert, and C. Hansen, "A real-time cloud modeling, rendering, and animation system," in *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2003, pp. 160–166.

[22] A. Elhaddad, F. Elhaddad, B. Sheng, S. Zhang, H. Sun, and E. Wu, "Real-time cloud simulation using lennard-jones approximation," in *Proceedings of the 29th International Conference on Computer Animation and Social Agents*, 2016, pp. 131–137.

[23] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[24] O. Ronneberger, P. Fischer, and T. 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.* Springer, 2015, pp. 234–241.

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

[26] Y. Alaluf, E. Richardson, G. Metzer, and D. Cohen-Or, "A neural space-time representation for text-to-image personalization," *ACM Transactions on Graphics (TOG)*, vol. 42, no. 6, pp. 1–10, 2023.

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

[28] J. Burgess, K.-C. Wang, and S. Yeung, "Viewpoint textual inversion: Unleashing novel view synthesis with pretrained 2d diffusion models," *arXiv preprint arXiv:2309.07986*, 2023.

[29] R. Liu, R. Wu, B. Van Hoorick, P. Tokmakov, S. Zakharov, and C. Vondrick, "Zero-1-to-3: Zero-shot one image to 3d object," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 9298–9309.

[30] blendermarket, "https://blendermarket.com/products/cloudscapes," 2023.

[31] K. Museth, "Vdb: High-resolution sparse volumes with dynamic topology," *ACM transactions on graphics (TOG)*, vol. 32, no. 3, pp. 1–22, 2013.

[32] blender, "https://www.blender.org/," 2023.

[33] polyhaven, "https://polyhaven.com/," 2023.

[34] N. Selin, "https://github.com/ophoperhpo/image-background-remove-tool," 2023.

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

[36] L. Melas-Kyriazi, I. Laina, C. Rupprecht, and A. Vedaldi, "Realfusion: 360deg reconstruction of any object from a single image," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 8446–8455.

[37] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, and R. Ng, "Fourier features let networks learn high frequency functions in low dimensional domains," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7537–7547, 2020.

[38] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[39] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.

[40] D. Vicini, S. Speierer, and W. Jakob, "Path replay backpropagation: differentiating light paths using constant memory and linear time," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, pp. 1–14, 2021.

[41] M. Nimier-David, D. Vicini, T. Zeltner, and W. Jakob, "Mitsuba 2: A retargetable forward and inverse renderer," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, pp. 1–17, 2019.

[42] K. Perlin, "Improving noise," in *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, 2002, pp. 681–682.

[43] S. Worley, "A cellular texture basis function," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996, pp. 291–294.

[44] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[45] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.