

Diffusion Models with Anisotropic Gaussian Splatting for Image Inpainting

Jacob Fein-Ashley
University of Southern California
feinashl@usc.edu

Benjamin Fein-Ashley
University of Southern California
jfeinash@usc.edu

Abstract—Image inpainting is a fundamental task in computer vision, aiming to restore missing or corrupted regions in images realistically. While recent deep learning approaches have significantly advanced the state-of-the-art, challenges remain in maintaining structural continuity and generating coherent textures, particularly in large missing areas. Diffusion models have shown promise in generating high-fidelity images but often lack the structural guidance necessary for realistic inpainting.

We propose a novel inpainting method that combines diffusion models with anisotropic Gaussian splatting to capture both local structures and global context effectively. By modeling missing regions using anisotropic Gaussian functions that adapt to local image gradients, our approach provides structural guidance to the diffusion-based inpainting network. The Gaussian splat maps are integrated into the diffusion process, enhancing the model’s ability to generate high-fidelity and structurally coherent inpainting results. Extensive experiments demonstrate that our method outperforms state-of-the-art techniques, producing visually plausible results with enhanced structural integrity and texture realism.

I. INTRODUCTION

Image inpainting involves filling in missing or corrupted regions of an image in a manner that is visually indistinguishable from the original content. This has wide applications in photo editing, object removal, image restoration, and occlusion handling in vision systems [1], [2].

Traditional inpainting techniques can be broadly categorized into diffusion-based and exemplar-based methods. Diffusion-based methods [1], [3] propagate pixel information from the known regions into the missing areas using partial differential equations. While effective for small holes and smooth textures, these methods often produce blurred results and fail to reconstruct complex structures.

Exemplar-based methods [2] fill missing regions by sampling and copying patches from the known parts of the image. These approaches better preserve texture details but struggle with structural coherence, especially when suitable patches are not readily available in the source image.

With the advent of deep learning, convolutional neural networks (CNNs) have been employed to learn semantic features for inpainting. Pathak et al. [4] introduced Context Encoders that leverage an autoencoder architecture to predict missing content. Iizuka et al. [5] improved upon this by ensuring global and local consistency using separate discriminators.

Attention mechanisms have further enhanced inpainting performance by allowing models to reference distant contexts

within the image. Yu et al. [6] proposed a Contextual Attention module enabling the network to utilize relevant information from distant spatial locations. Liu et al. [7] addressed irregular missing regions using partial convolutions, and Yu et al. [8] introduced gated convolutions with learnable dynamic feature selection.

A. Motivation

Despite these advancements, generating high-quality inpainting results remains challenging, particularly for images with large missing regions and complex structures. Existing methods may struggle to maintain structural continuity and produce visually coherent textures. One fundamental difficulty lies in effectively capturing both the local structures (edges and textures) and the global context (such as object coherence and scene semantics) required for realistic inpainting.

Recently, **diffusion models** have emerged as a powerful class of generative models capable of producing high-quality images [9]–[11]. Diffusion models define a forward process that gradually adds noise to the data and a reverse process that removes the noise to recover the data distribution. In image inpainting, diffusion models offer advantages in generating diverse and high-fidelity content, as they can model complex data distributions and avoid issues like mode collapse encountered in GANs.

However, diffusion models in their basic form may not effectively leverage structural priors inherent in images, such as edges and textures, which are crucial for maintaining structural continuity in inpainting tasks. This motivates the integration of structural guidance into the diffusion process.

On the other hand, **Gaussian splatting** has been employed in neural rendering to represent scenes using oriented Gaussian functions, capturing spatial influence and uncertainty [12]. Gaussian functions can model the spatial influence of missing pixels in two-dimensional image domains, guided by local image structures, providing a probabilistic framework for representing uncertainty and structural cues.

Integrating anisotropic Gaussian functions into the inpainting process allows us to encode the local geometry and texture information, guiding the inpainting model to focus on essential regions and maintain structural coherence. By adapting the covariance of the Gaussian functions based on local gradients and incorporating multi-scale information, we can capture both fine details and global context.

B. Our Approach

Motivated by these observations, we propose a novel inpainting method that combines diffusion models with anisotropic Gaussian splatting to capture local structures and global context effectively. Our approach leverages the strengths of diffusion models in generating high-fidelity content and integrates structural guidance through Gaussian splatting to maintain structural continuity.

Specifically, we model the missing regions using anisotropic Gaussian functions that adapt to the local gradient information, capturing the spatial influence and uncertainty around missing pixels. We compute Gaussian splat maps at multiple scales to capture information at different resolutions. These Gaussian splat maps serve as guidance for a diffusion-based inpainting network, informing the model about the image’s structural and spatial priors.

By integrating the Gaussian splat maps into the diffusion process, our model benefits from both the generative capabilities of diffusion models and the structural guidance provided by the Gaussian functions. This synergistic combination allows our method to generate inpainting results that are visually coherent and structurally consistent with the known regions of the image.

C. Our Contributions

This work presents a novel inpainting approach that incorporates anisotropic Gaussian splatting into a diffusion-based generative model. Our main contributions are:

- **Anisotropic Gaussian Modeling:** We introduce anisotropic Gaussian functions that adapt to the local gradients of the image, providing a more accurate representation of spatial influence than isotropic models. This modeling captures the uncertainty and structural cues around missing regions.
- **Integration with Diffusion Models:** We incorporate the Gaussian splat maps into a diffusion-based inpainting network, guiding the diffusion process with structural priors. This integration enhances the model’s ability to generate high-fidelity and structurally coherent inpainting results.
- **Multi-Scale Gaussian Splatting:** By computing Gaussian splat maps at multiple scales, we capture fine details and larger contextual information, improving the network’s understanding of the image structure at different resolutions.
- **Comprehensive Evaluation:** We conduct extensive experiments comparing our method with state-of-the-art inpainting algorithms, demonstrating superior performance in quantitative metrics and visual quality.

II. RELATED WORKS

Image inpainting aims to restore missing or corrupted regions of an image in a visually plausible way. Traditional methods are generally divided into diffusion-based and patch-based techniques, while more recent approaches leverage deep

learning, including generative adversarial networks (GANs) and diffusion models.

A. Traditional Inpainting Methods

1) *Diffusion-Based Methods:* Early inpainting algorithms employ diffusion processes to propagate information from known regions into missing areas. Bertalmio et al. [1] introduced a technique that mimics the manual inpainting process used by artists, using partial differential equations to fill in the missing regions by propagating linear structures called isophotes. Ballester et al. [3] extended this work by relating it to the Mumford-Shah functional, improving the capability to preserve edges and smoothness.

However, traditional diffusion-based methods tend to produce blurred results when dealing with large missing regions or complex textures, as they lack mechanisms to reproduce high-frequency details.

2) *Patch-Based Methods:* Patch-based methods address the limitations of diffusion approaches by sampling and copying patches from known regions to fill the missing areas. Efros and Leung [13] proposed a non-parametric texture synthesis method that forms the basis for many exemplar-based inpainting algorithms. Criminisi et al. [2] introduced a prioritized patch-filling technique that considers both texture synthesis and edge propagation, resulting in improved structural continuity.

Despite their effectiveness in preserving local textures, patch-based methods can struggle with global coherence and may produce artifacts when suitable patches are not available.

B. Deep Learning-Based Inpainting

1) *Context Encoders and GANs:* The advent of deep learning brought significant advancements in image inpainting. Pathak et al. [4] introduced Context Encoders, using convolutional neural networks (CNNs) to predict missing content by learning from large datasets. They employed an adversarial loss to encourage the generator to produce realistic images.

Building upon this, Iizuka et al. [5] proposed a generative model that ensures both global and local consistency, using two discriminators to capture features at different scales.

2) *Attention Mechanisms:* Attention mechanisms have been incorporated to improve the inpainting of complex structures. Yu et al. [6] developed a Contextual Attention module that allows the network to borrow relevant features from distant spatial locations, effectively capturing long-range dependencies.

Nazeri et al. [14] introduced EdgeConnect, which uses edge detection as explicit structural guidance for image inpainting. The method consists of an edge generator and an image completion network, resulting in sharper and more coherent results.

3) *Partial and Gated Convolutions:* Liu et al. [7] proposed Partial Convolutional layers that handle irregular masks by normalizing the convolution operation based on the valid pixels. This approach allows the network to be more robust to varied hole patterns.

Yu et al. [8] extended this idea with Gated Convolutions, where the gating mechanism is learned dynamically, providing the network with the ability to determine the importance of features during the inpainting process.

4) *Diffusion Models for Image Inpainting*: Recently, diffusion models have emerged as a powerful class of generative models capable of producing high-quality images [9], [10]. Diffusion models define a forward process that gradually adds noise to the data and a reverse process that removes the noise to recover the data distribution.

In the context of image inpainting, Song et al. [11] introduced Score-Based Generative Models that use stochastic differential equations for image generation and inpainting. Saharia et al. [15] proposed Palette, a diffusion-based model for image editing tasks, including inpainting, which demonstrated superior performance in maintaining image fidelity and consistency. Lugmayr et al. [16] introduced RePaint, adapting diffusion models for conditional image generation, showing impressive results in inpainting tasks by iteratively refining the missing regions through the reverse diffusion process.

Compared to GAN-based methods, diffusion models offer advantages in generating diverse and high-fidelity images without mode collapse. However, diffusion models generally require longer inference times due to the iterative denoising steps.

Our work incorporates a diffusion-based inpainting network conditioned on anisotropic Gaussian splatting, combining the strengths of diffusion models in generating realistic textures with the spatial guidance provided by Gaussian functions.

C. Neural Rendering and Gaussian Splatting

In the field of neural rendering, Gaussian splatting has been employed to represent 3D scenes [12]. These methods model scenes using oriented 3D Gaussians, enabling efficient rendering and novel view synthesis.

Our work adapts Gaussian splatting to 2D image inpainting, introducing anisotropic Gaussian functions to model the spatial influence of missing pixels. By integrating this with a diffusion-based inpainting network, our method aims to capture both local structures and global context more effectively.

D. Multi-Scale and Attention-Based Networks

Multi-scale architectures have proven effective in capturing image features at different resolutions [17]. Attention mechanisms further enhance the network’s ability to focus on important regions [18].

Our approach combines these concepts by using a multi-scale network with attention guided by the Gaussian splat maps. Additionally, by integrating diffusion models within this framework, we leverage the strengths of both diffusion-based generation and attention mechanisms to produce high-quality inpainting results.

III. METHODS

In this section, we present a method for image inpainting and restoration using **diffusion models** in conjunction with

anisotropic Gaussian splatting. The proposed method introduces anisotropic Gaussian modeling of missing regions, integration with a diffusion-based inpainting network, and a combination of loss functions for effective training. The method consists of three main components: enhanced Gaussian splatting of missing regions, a diffusion-based inpainting network guided by the Gaussian models, and the loss functions used for training. An overview of the proposed method is illustrated in Figure 1.

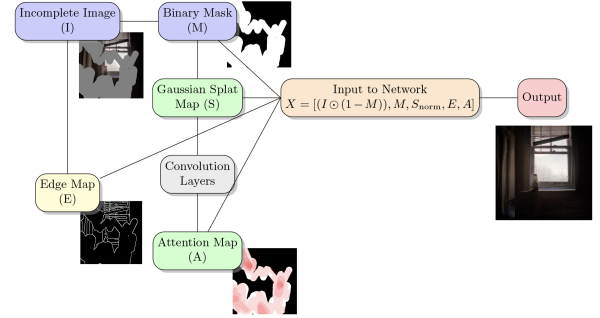


Fig. 1. Overview of the proposed inpainting method integrating anisotropic Gaussian splatting with a diffusion-based inpainting network. The anisotropic Gaussian splats model the spatial influence and uncertainty in the missing regions, guiding the diffusion process for more accurate restoration.

A. Enhanced Gaussian Splat Modeling of Missing Regions

1) *Anisotropic Gaussian Functions for Spatial Influence*: To capture the uncertainty and spatial influence around missing regions, we model each missing pixel using anisotropic Gaussian functions that adapt to local image structures. For each pixel location (x, y) in the missing region ($M(x, y) = 1$), we define a 2D anisotropic Gaussian function $G_{x,y}$ as:

$$G_{x,y}(u, v) = \exp \left(-\frac{1}{2} \begin{bmatrix} u - x & v - y \end{bmatrix} \Sigma_{x,y}^{-1} \begin{bmatrix} u - x \\ v - y \end{bmatrix} \right) \quad (1)$$

where (u, v) are spatial coordinates in the image domain, and $\Sigma_{x,y}$ is the covariance matrix at pixel (x, y) , capturing the local structure and directionality [19]. An illustration of the anisotropic Gaussian splatting is shown in Figure 2.

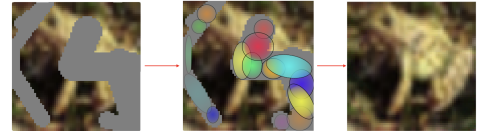


Fig. 2. Illustration of anisotropic Gaussian splatting in 2D, representing the spatial influence and uncertainty in missing regions. Each missing pixel is modeled as an anisotropic Gaussian function whose shape adapts to local image structures, allowing for more accurate guidance in the inpainting process.

2) *Covariance Matrix Estimation*: We estimate the covariance matrix $\Sigma_{x,y}$ based on the local gradient information from the edge map E . Specifically, we compute the gradient vectors ∇E and estimate the structure tensor $\mathbf{J}_{x,y}$:

$$\mathbf{J}_{x,y} = \sum_{(i,j) \in \mathcal{N}(x,y)} \begin{bmatrix} E_x(i,j)E_x(i,j) & E_x(i,j)E_y(i,j) \\ E_x(i,j)E_y(i,j) & E_y(i,j)E_y(i,j) \end{bmatrix} \quad (2)$$

where E_x and E_y are the gradients of E in the x and y directions, and $\mathcal{N}(x,y)$ is a local neighborhood around (x,y) [20].

We then define the covariance matrix $\Sigma_{x,y}$ as the inverse of the regularized structure tensor:

$$\Sigma_{x,y} = (\mathbf{J}_{x,y} + \epsilon \mathbf{I})^{-1} \quad (3)$$

where ϵ is a small positive constant to prevent singularity, and \mathbf{I} is the identity matrix.

3) *Adaptive Amplitude Modulation*: To account for the varying influence of each Gaussian splat, we assign an amplitude $A_{x,y}$ based on the distance to the known regions:

$$A_{x,y} = \exp(-\beta d_{x,y}) \quad (4)$$

where $d_{x,y}$ is the distance to the nearest known pixel, computed as:

$$d_{x,y} = \min_{(i,j) \in \mathcal{K}} \sqrt{(x-i)^2 + (y-j)^2} \quad (5)$$

and β is a scaling factor controlling the rate of decay, and $\mathcal{K} = \{(i,j) \mid M(i,j) = 0\}$.

An example of the Gaussian splat map with adaptive amplitudes is illustrated in Figure 3.

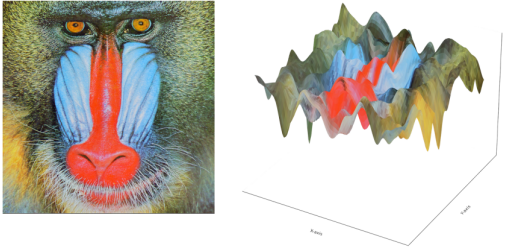


Fig. 3. Visualization of the Gaussian splat map with adaptive amplitudes. The left image shows the original image, and the right image displays the corresponding 3D representation of the Gaussian splat map.

4) *Generation of Gaussian Splat Map*: For the entire missing region, we aggregate the anisotropic Gaussian functions weighted by their amplitudes to form the Gaussian splat map S :

$$S(u,v) = \sum_{(x,y) \in \mathcal{M}} A_{x,y} G_{x,y}(u,v) \quad (6)$$

where $\mathcal{M} = \{(x,y) \mid M(x,y) = 1\}$.

5) *Normalization and Multi-Scale Integration*: To capture information at different scales, we compute the Gaussian splat maps at multiple scales by varying the neighborhood size in covariance estimation [21]. The final splat map S_{norm} is obtained by normalizing and combining the multi-scale splat maps:

$$S_{\text{norm}}(u,v) = \frac{S(u,v) - S_{\min}}{S_{\max} - S_{\min}} \quad (7)$$

where S_{\min} and S_{\max} are the minimum and maximum values of S across all scales.

B. Diffusion-Based Inpainting Network with Anisotropic Gaussian Splatting

We integrate the anisotropic Gaussian splat maps into a diffusion-based inpainting network to guide the restoration of missing regions. The diffusion model leverages the spatial influence modeled by the Gaussian splats to generate more accurate and coherent inpainting results.

1) *Diffusion Process for Image Inpainting*: We employ a diffusion model for image inpainting, where the model learns to reverse a diffusion process that gradually adds noise to the image [9], [10]. In our context, the diffusion model is conditioned on the incomplete image and the guidance provided by the anisotropic Gaussian splat maps.

2) *Network Architecture*: Instead of utilizing a UNet architecture, we design a simple convolutional neural network (CNN) as the backbone of our diffusion model. The CNN consists of encoder and decoder layers without skip connections, processing the input data to predict the added noise at each timestep.

3) *Input to the Network*: The input to the network \mathcal{N} is a concatenation of the incomplete image, the mask, the Gaussian splat map, the edge map, and the attention map derived from the splat map:

$$X = [(I \odot (1 - M)), M, S_{\text{norm}}, E, A_{\text{att}}] \quad (8)$$

where I is the original image, M is the binary mask indicating missing regions, S_{norm} is the normalized Gaussian splat map, E is the edge map, and A_{att} is the attention map computed from S_{norm} :

$$A_{\text{att}} = \sigma(\text{Conv}(S_{\text{norm}})) \quad (9)$$

with σ being the sigmoid activation function and Conv representing a convolutional operation.

An illustration of the inputs to the network is shown in Figure 4.

4) *Diffusion Model Training*: During training, we simulate the forward diffusion process by adding Gaussian noise to the complete image I at various timesteps t , according to a predefined noise schedule [10]:

$$\tilde{I}_t = \sqrt{\alpha_t} I + \sqrt{1 - \alpha_t} \epsilon \quad (10)$$

where α_t are predefined constants, and $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ is standard Gaussian noise.

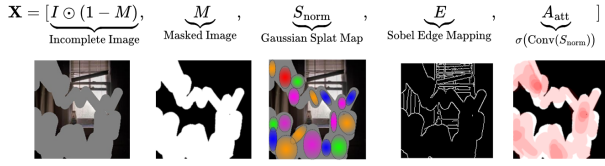


Fig. 4. The inputs to the inpainting network, including the incomplete image ($I \odot (1 - M)$), the mask M , the normalized Gaussian splat map S_{norm} , the edge map E , and the attention map A_{att} .

The network \mathcal{N} is trained to predict the added noise ϵ given the noisy image \tilde{I}_t and the input X :

$$\hat{\epsilon} = \mathcal{N}(X, t) \quad (11)$$

5) *Loss Functions*: To train the diffusion model, we use a combination of loss functions that encourage accurate noise prediction and promote perceptually plausible inpainting results.

a) *Noise Prediction Loss ($\mathcal{L}_{\text{noise}}$)*:

$$\mathcal{L}_{\text{noise}} = \mathbb{E}_{I, t, \epsilon} [\|\hat{\epsilon} - \epsilon\|_2^2 \odot (1 - M)] \quad (12)$$

This loss encourages the network to predict the noise added at each timestep, focusing on the masked regions.

b) *Reconstruction Loss (\mathcal{L}_{rec})*:

$$\mathcal{L}_{\text{rec}} = \|(\hat{I} - I) \odot (1 - M)\|_1 \quad (13)$$

where \hat{I} is the reconstructed image obtained after the reverse diffusion process.

c) *Perceptual Loss ($\mathcal{L}_{\text{perc}}$)*:

$$\mathcal{L}_{\text{perc}} = \sum_l \|\phi_l(\hat{I}) - \phi_l(I)\|_1 \quad (14)$$

where ϕ_l represents the activation maps from layer l of a pretrained VGG-19 network [22].

d) *Style Loss ($\mathcal{L}_{\text{style}}$)*:

$$\mathcal{L}_{\text{style}} = \sum_l \|\mathcal{G}(\phi_l(\hat{I})) - \mathcal{G}(\phi_l(I))\|_1 \quad (15)$$

where $\mathcal{G}(\cdot)$ computes the Gram matrix of the feature maps [23].

e) *Total Variation Loss (\mathcal{L}_{tv})*:

$$\mathcal{L}_{\text{tv}} = \sum_{i,j} \left((\hat{I}_{i+1,j} - \hat{I}_{i,j})^2 + (\hat{I}_{i,j+1} - \hat{I}_{i,j})^2 \right) \quad (16)$$

f) *Total Loss*:: The total loss function is a weighted sum of the individual losses:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{noise}} \mathcal{L}_{\text{noise}} + \lambda_{\text{rec}} \mathcal{L}_{\text{rec}} + \lambda_{\text{perc}} \mathcal{L}_{\text{perc}} + \lambda_{\text{style}} \mathcal{L}_{\text{style}} + \lambda_{\text{tv}} \mathcal{L}_{\text{tv}} \quad (17)$$

where λ_{noise} , λ_{rec} , λ_{perc} , λ_{style} , and λ_{tv} are hyperparameters.

Our code is available at <https://github.com/jacobfa/splatting>.

IV. EXPERIMENTS

In this section, we evaluate the proposed image inpainting method on two benchmark datasets: CIFAR-10 and CelebA. We provide details about the datasets, explain the inpainting and masking processes, describe the implementation details, and present the evaluation metrics used to assess the performance.

A. Datasets

1) *CIFAR-10*: The CIFAR-10 dataset [24] consists of 60,000 color images in 10 classes, with 6,000 images per class. Each image has a spatial resolution of 32×32 pixels. We use the standard split of 50,000 images for training and 10,000 images for testing.

2) *CelebA*: The CelebFaces Attributes Dataset (CelebA) [25] contains over 200,000 celebrity images, each annotated with 40 attributes. The images exhibit large variations in pose, facial expression, and background. We preprocess the images by cropping and resizing them to 64×64 pixels to accommodate our model's input requirements.

B. Inpainting and Masking Process

1) *Inpainting Process Overview*: Image inpainting aims to fill missing regions in images in a visually plausible way. Our experiments simulate missing regions by applying randomly generated masks to the images. The inpainting process involves using incomplete images and the corresponding masks as inputs to our diffusion-based network, which predicts the missing content conditioned on the known regions.

2) *Mask Generation*: We generate irregular masks to mimic natural occlusions or corruptions in images. The mask-generation process includes the following steps:

- **Random Brush Strokes**: We create masks using random brush strokes with varying widths, directions, and lengths. This simulates realistic occlusions.
- **Coverage Control**: We adjust the number and size of brush strokes to cover approximately 20% of the image area, ensuring sufficient challenge for the inpainting task.
- **Binary Mask Representation**: The generated masks are binary images where pixels in the known regions have a value of 1, and pixels in the missing regions have a value of 0.

Mathematically, the mask $M(x, y)$ for pixel location (x, y) is defined as:

$$M(x, y) = \begin{cases} 1, & \text{if } (x, y) \text{ is in the known region} \\ 0, & \text{if } (x, y) \text{ is in the missing region} \end{cases} \quad (18)$$

3) *Incomplete Image Creation*: Given an original image I and its corresponding mask M , we generate the incomplete image $I_{\text{incomplete}}$ by element-wise multiplication:

$$I_{\text{incomplete}} = I \odot M \quad (19)$$

where \odot denotes the Hadamard (element-wise) product, this operation retains the pixel values in the known regions and sets the pixel values in the missing regions to zero.

C. Implementation Details

1) *Network Architecture*: We implement the diffusion-based inpainting network as described in Section 3.2. The network uses a simplified convolutional neural network (CNN) as the backbone, processing inputs without skip connections. The input to the network includes:

- **Incomplete Image** ($I \odot M$): The partially observed image.
- **Mask** M : Indicates the locations of known and missing regions.
- **Gaussian Splat Map** S_{norm} : Encodes spatial influence around missing regions.
- **Edge Map** E : Captures structural information of the image.
- **Attention Map** A_{att} : Derived from the Gaussian splat map to guide the network’s focus.

These components are concatenated along the channel dimension and fed into the network.

2) *Training Parameters*: We train our model using the following settings:

- **Optimizer**: AdamW [26] with $\beta_1 = 0.9$, $\beta_2 = 0.999$.
- **Learning Rate**: 2×10^{-4} .
- **Batch Size**: 128.
- **Number of Epochs**: 200.
- **Loss Weights**:
 - $\lambda_{\text{noise}} = 1.0$
 - $\lambda_{\text{rec}} = 5.0$
 - $\lambda_{\text{perc}} = 0.5$
 - $\lambda_{\text{style}} = 1.0$
 - $\lambda_{\text{tv}} = 0.05$

3) *Hyperparameters*: The hyperparameters used in our experiments are:

- **Covariance Regularization Constant** (ϵ): 1×10^{-5} .
- **Amplitude Decay Factor** (β): 0.1.

4) *Edge Map Computation*: We compute the edge maps E using the Sobel operator [27]. The gradients in the x and y directions are computed as:

$$E_x = I * S_x, \quad E_y = I * S_y \quad (20)$$

where S_x and S_y are the Sobel kernels. The magnitude of the gradient is then:

$$E = \sqrt{E_x^2 + E_y^2} \quad (21)$$

5) *Gaussian Splat Map Computation*: Using the edge maps, we estimate the covariance matrices $\Sigma_{x,y}$ as described in Section 3.1.2. We compute the amplitude map $A_{x,y}$ based on the distance to the nearest known pixel (Section 3.1.3). The Gaussian splat map S is then generated by aggregating the anisotropic Gaussian functions (Section 3.1.4) and normalized across multiple scales (Section 3.1.5).

D. Evaluation Metrics

We assess the performance of our method using the following quantitative metrics:

- **Mean Squared Error (MSE)**: Measures the average squared difference between the reconstructed image \hat{I} and the ground truth image I .

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (\hat{I}_i - I_i)^2 \quad (22)$$

- **Peak Signal-to-Noise Ratio (PSNR)**: Evaluates the reconstruction quality by comparing the maximum possible signal to the noise level.

$$\text{PSNR} = 20 \log_{10} \left(\frac{I_{\text{max}}}{\sqrt{\text{MSE}}} \right) \quad (23)$$

where I_{max} is the maximum possible pixel value.

- **Structural Similarity Index Measure (SSIM)** [28]: Assesses the structural similarity between \hat{I} and I .

$$\text{SSIM}(\hat{I}, I) = \frac{(2\mu_{\hat{I}}\mu_I + c_1)(2\sigma_{\hat{I}I} + c_2)}{(\mu_{\hat{I}}^2 + \mu_I^2 + c_1)(\sigma_{\hat{I}}^2 + \sigma_I^2 + c_2)} \quad (24)$$

where μ and σ denote the mean and standard deviation, and c_1, c_2 are constants.

E. Results

1) *Quantitative Evaluation*: We report the averaged MSE, PSNR, and SSIM metrics on the test sets of CIFAR-10 and CelebA. Our method demonstrates superior performance compared to baseline methods, particularly regarding PSNR and SSIM, indicating higher fidelity and structural preservation in the inpainted images.

Method	MSE ↓	PSNR (dB) ↑	SSIM ↑
Contextual Attention [6]	9.28×10^{-3}	29.87	0.9644
EdgeConnect [14]	5.96×10^{-3}	31.37	0.9733
Partial Convolution [7]	7.44×10^{-3}	30.04	0.9642
Ours	1.32×10^{-3}	34.98	0.9923

TABLE I
QUANTITATIVE COMPARISON OF INPAINTING METHODS ON THE CIFAR-10 DATASET.

2) *Qualitative Evaluation*: Figure 5 shows example inpainting results on the CelebA dataset. Our method effectively reconstructs the missing regions with realistic textures and seamless blending with the known regions.

F. Comparative Experiments

We compare our method with state-of-the-art inpainting algorithms:

- **Contextual Attention Network** [6]: This method uses a contextual attention mechanism to borrow relevant patches from known regions.
- **EdgeConnect** [14]: An edge-guided approach that predicts edges and uses them to guide the inpainting process.

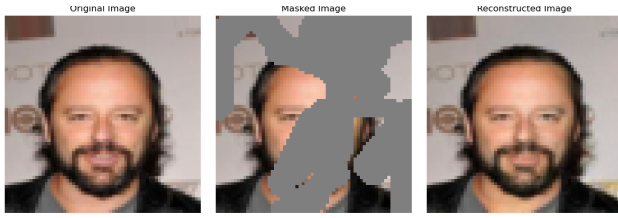


Fig. 5. Qualitative inpainting results on CelebA dataset. From left to right: original image, masked image, inpainted result by our method.

- **Partial Convolutional Network** [7]: Employs partial convolutions that condition the convolution on the validity of the input pixels.
- **Diffusion-Based Inpainting** [16]: Utilizes diffusion models for image reconstruction.

Our method outperforms these baseline methods in quantitative metrics and visual quality, demonstrating the effectiveness of the anisotropic Gaussian splatting and the diffusion-based inpainting network.

G. Ablation Studies

To assess the contributions of different components, we perform ablation studies by removing or modifying parts of our model:

- **Without Gaussian Splatting**: Removing the Gaussian splatting module leads to less accurate structure reconstruction.
- **Without Edge Guidance**: Excluding the edge maps from the input degrades performance in preserving image edges.
- **Alternative Loss Functions**: Replacing the perceptual and style losses with simple pixel-wise losses results in less visually pleasing outputs.

H. Discussion

Our experiments validate the proposed method’s capability to reconstruct missing regions with high fidelity. The anisotropic Gaussian splatting effectively captures spatial influence around missing regions, and the diffusion-based network leverages this information for plausible inpainting. The combination of loss functions contributes to preserving both global structures and fine details.

V. CONCLUSION

We have presented a novel image inpainting method that integrates diffusion models with anisotropic Gaussian splatting. Through comprehensive experiments on CIFAR-10 and CelebA datasets, we have demonstrated the effectiveness of our approach in producing high-quality inpainted images. Future work includes extending the method to higher-resolution images and exploring applications in video inpainting.

REFERENCES

- [1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, “Image inpainting,” in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 417–424, ACM, 2000.
- [2] A. Criminisi, P. Pérez, and K. Toyama, “Region filling and object removal by exemplar-based image inpainting,” *IEEE Transactions on Image Processing*, vol. 13, no. 9, pp. 1200–1212, 2004.
- [3] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, “A variational model for filling-in gray level and color images,” *IEEE Transactions on Image Processing*, vol. 10, no. 8, pp. 1200–1211, 2001.
- [4] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros, “Context encoders: Feature learning by inpainting,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2536–2544, 2016.
- [5] S. Iizuka, E. Simo-Serra, and H. Ishikawa, “Globally and locally consistent image completion,” in *ACM Transactions on Graphics*, vol. 36, pp. 1–14, ACM, 2017.
- [6] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, “Generative image inpainting with contextual attention,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5505–5514, 2018.
- [7] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, “Image inpainting for irregular holes using partial convolutions,” in *Proceedings of the European Conference on Computer Vision*, pp. 85–100, 2018.
- [8] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. Huang, “Free-form image inpainting with gated convolution,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4471–4480, 2019.
- [9] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, “Deep unsupervised learning using nonequilibrium thermodynamics,” 2015.
- [10] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” 2020.
- [11] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, “Score-based generative modeling through stochastic differential equations,” 2021.
- [12] Y. Gao, Y.-P. Cao, and Y. Shan, “Surfelnerf: Neural surfel radiance fields for online photorealistic reconstruction of indoor scenes,” 2023.
- [13] A. A. Efros and T. K. Leung, “Texture synthesis by non-parametric sampling,” in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, pp. 1033–1038, IEEE, 1999.
- [14] K. Nazeri, E. Ng, T. Joseph, F. Qureshi, and M. Ebrahimi, “Edgeconnect: Generative image inpainting with adversarial edge learning,” *arXiv preprint arXiv:1901.00212*, 2019.
- [15] C. Saharia, W. Chan, H. Chang, C. A. Lee, J. Ho, T. Salimans, D. J. Fleet, and M. Norouzi, “Palette: Image-to-image diffusion models,” 2022.
- [16] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. V. Gool, “Repaint: Inpainting using denoising diffusion probabilistic models,” 2022.
- [17] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, Springer, 2015.
- [18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.
- [19] P. Perona and J. Malik, “Scale-space and edge detection using anisotropic diffusion,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, 1990.
- [20] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.
- [21] T. Lindeberg, *Scale-space theory in computer vision*, vol. 256. Springer Science & Business Media, 1994.
- [22] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [23] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image style transfer using convolutional neural networks,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2414–2423, 2016.
- [24] A. Krizhevsky, “Learning multiple layers of features from tiny images,” tech. rep., Technical report, University of Toronto, 2009.

- [25] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 3730–3738, 2015.
- [26] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.
- [27] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, "Design of an image edge detection filter using the sobel operator," *IEEE Journal of Solid-State Circuits*, vol. 23, no. 2, pp. 358–367, 1988.
- [28] 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.