

2D Gaussian Splatting with Semantic Alignment for Image Inpainting

Hongyu Li¹, Chaofeng Chen², Xiaoming Li³, Guangming Lu¹

¹Harbin Institute of Technology, Shenzhen

²School of Artificial Intelligence, Wuhan University

³Nanyang Technological University

<https://github.com/hitlhy715/2DGS-inpaint>

Abstract

Gaussian Splatting (GS), a recent technique for converting discrete points into continuous spatial representations, has shown promising results in 3D scene modeling and 2D image super-resolution. In this paper, we explore its untapped potential for image inpainting, which demands both locally coherent pixel synthesis and globally consistent semantic restoration. We propose the first image inpainting framework based on 2D Gaussian Splatting, which encodes incomplete images into a continuous field of 2D Gaussian splat coefficients and reconstructs the final image via a differentiable rasterization process. The continuous rendering paradigm of GS inherently promotes pixel-level coherence in the inpainted results. To improve efficiency and scalability, we introduce a patch-wise rasterization strategy that reduces memory overhead and accelerates inference. For global semantic consistency, we incorporate features from a pretrained DINO model. We observe that DINO's global features are naturally robust to small missing regions and can be effectively adapted to guide semantic alignment in large-mask scenarios, ensuring that the inpainted content remains contextually consistent with the surrounding scene. Extensive experiments on standard benchmarks demonstrate that our method achieves competitive performance in both quantitative metrics and perceptual quality, establishing a new direction for applying Gaussian Splatting to 2D image processing.

1 Introduction

Human visual perception naturally interprets the world as a continuous experience. In contrast, digital images are constrained by hardware and storage limitations, resulting in representations composed of discrete pixels. Similarly, prevailing neural network architectures, such as Convolutional Neural Networks (CNNs) and Transformers, operate on pixel-based inputs and rely on fixed spatial encodings. This fundamental mismatch limits the ability of networks to fully capture the continuous nature of real-world visual data, particularly in the spatial domain, thereby restricting their expressive capacity and representational fidelity.

To address this challenge, recent research has increasingly focused on integrating the complementary strengths of discrete and continuous representations (Tian et al. 2023; Jiao et al. 2025; Jiang et al. 2024). Implicit Neural Representations (INRs) (Chen, Liu, and Wang 2021; Cao et al. 2023; Wu, Ni, and Zhang 2023; Li et al. 2022a) have made signif-

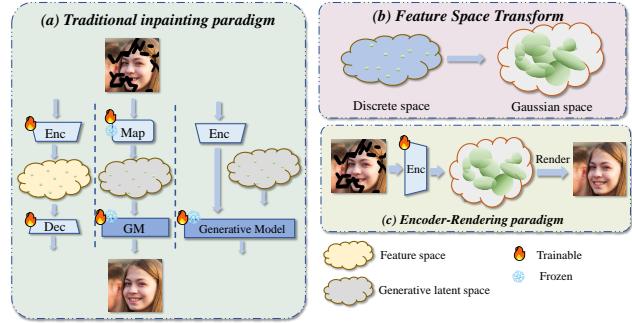


Figure 1: (a) Traditional inpainting methods rely on CNNs or Transformers to synthesize missing pixels in a spatially discrete manner. (b) Gaussian splatting generates pixels in a continuous way. (c) Our method introduces a novel paradigm: instead of synthesizing pixel values directly, we encode the input image into a learned Gaussian feature space and reconstruct it through a differentiable rasterization process, allowing for continuous and smoother pixel generation.

icant strides by learning mappings from spatial coordinates to RGB values, enabling finer spatial granularity and continuous modeling. Latest methods (Peng et al. 2025; Hu et al. 2025) employ 2D Gaussian Splatting to naturally encode local continuous features at arbitrary scales.

Meanwhile, the potential of Gaussian Splatting (GS) for image inpainting remains underexplored. GS represents images as continuous fields using localized, overlapping Gaussians, enabling smooth interpolation and fine-grained detail reconstruction. These properties make it well-suited for filling in missing regions with both spatial continuity and semantic coherence. In contrast, as illustrated in Figure 1(a), most existing inpainting methods rely on CNNs or Transformers to synthesize missing pixels. However, their inherently discrete spatial processing often hinders the reconstruction of coherent pixel-level structures, particularly in regions with complex textures or fine details.

In this paper, we introduce a novel approach to image inpainting based on Gaussian splatting, positing the hypothesis that a complete Gaussian representation of an image can be modeled from its incomplete regions, leveraging the inherent continuity of Gaussian functions to restore high-quality images. Our framework maps incomplete images to

dense Gaussian feature fields, which are then rendered back to the pixel domain to produce complete images, as illustrated in Figure 1(c).

To address the computational challenges of high-resolution inpainting, we introduce a patch-level rasterization strategy. This approach processes images in smaller, manageable segments, significantly reducing GPU memory demands and computational overhead by exploiting the local coherence inherent in image patches. While this patch-based processing improves efficiency, it introduces challenges in maintaining global semantic consistency across patches.

To ensure global semantic coherence, we investigate the robustness of DINO features for inpainting tasks. We find that DINO features demonstrate remarkable robustness to small masks and can be effectively adapted to handle large masks through simple adaptation techniques. Building on this observation, we propose incorporating DINO features as global semantic guidance within our Gaussian splatting framework. This integration enables the model to maintain semantic consistency across patches while preserving the computational benefits of our patch-level approach.

The main contributions of this paper are:

- **A novel 2D Gaussian Splatting framework for image inpainting.** To the best of our knowledge, this is the first approach to leverage 2D Gaussian splatting for high-quality image inpainting. Our method directly maps known image pixels into a continuous Gaussian feature space, enabling effective reconstruction of missing regions through the inherent continuity of Gaussian functions without requiring explicit optimization of scene-wide parameters from scratch.
- **Patch-level rasterization for scalable high-resolution processing.** We introduce a patch-wise processing strategy that addresses the computational challenges of high-resolution inpainting by dividing images into manageable segments. This approach significantly reduces GPU memory consumption and accelerates rendering while incorporating overlapping regions and blending techniques to maintain spatial continuity across patch boundaries.
- **DINO feature adaptation for global semantic guidance.** We demonstrate that DINO features exhibit remarkable robustness to small masks and can be effectively adapted to handle large masks through simple techniques. Leveraging this discovery, we integrate DINO features as global semantic guidance via Adaptive Layer Normalization (AdaLN), enabling the model to maintain semantic consistency across patches while conditioning representations on high-level contextual cues for accurate reconstruction of complex structures and object relationships.

2 Related Work

2.1 Gaussian Splatting

3D Gaussian Splatting 3D Gaussian Splatting (3DGS) (Kerbl et al. 2023) has recently emerged as a compelling alternative paradigm in computer graphics, enabling 3D scene synthesis via explicit, learnable Gaussian representations. Unlike implicit neural representations (INRs) such as

NeRF (Mildenhall et al. 2021), which rely on coordinate-based function mappings, 3DGS represents scenes using a large number of parameterized Gaussians combined with differentiable rasterization techniques, supporting real-time rendering and fine-grained editability. This approach has shown remarkable effectiveness in various applications (Keetha et al. 2024; Luiten et al. 2024; Chen et al. 2024; Huang et al. 2024).

2D Gaussian Splatting Building upon 3DGS, 2D Gaussian Splatting (2DGS) exploits similar principles in image processing, benefiting from favorable mathematical properties for representing continuous image information. Recently, 2DGS has demonstrated success in tasks such as image tokenization and super-resolution. Methodologically, existing works fall into two categories: 1) Using Gaussians to diffuse or broadcast image features. For example, GaussianSR (Hu et al. 2025) constructs a continuous feature space for Arbitrary-Scale Super-Resolution (ASSR) by diffusing pixel-level features into a higher-resolution space. Similarly, GaussianToken (Dong et al. 2025) uses 2DGS to overcome the limited representation capacity caused by discrete feature spaces in vector quantization methods (Esser, Rombach, and Ommer 2021). 2) Direct modeling of Gaussian spaces. GaussianImage (Zhang et al. 2024) pioneers the use of 2DGS for image compression by optimizing Gaussian parameters, although their method currently processes only one image at a time. GSASR (Chen et al. 2025) proposes a feature injection module that constructs a 2D Gaussian space via learnable embeddings. PixeltoGaussian (Peng et al. 2025) empirically analyzes Gaussian space properties, deriving priors that improve early-stage convergence speed and training stability. To the best of our knowledge, however, the potential of 2D Gaussian Splatting for image inpainting remains unexplored.

2.2 Image Inpainting

Image inpainting is a long-standing research problem that has undergone substantial evolution from early exemplar-based methods (Efros and Leung 1999; He and Sun 2014) to modern learning-based approaches (Pathak et al. 2016; Liu et al. 2018; Yu et al. 2018; Guo, Yang, and Huang 2021). The introduction of perceptual loss (Johnson, Alahi, and Fei-Fei 2016) and adversarial training (Goodfellow et al. 2014) has further propelled the field by enabling models to generate visually plausible and semantically coherent content.

Convolutional neural networks (CNNs) have been widely adopted for inpainting (Liu et al. 2018; Yu et al. 2019; Zheng, Cham, and Cai 2019). While effective, CNN-based models are fundamentally limited by their restricted receptive fields. To address this, recent approaches have turned to transformer architectures for their superior capacity to model long-range dependencies (Li et al. 2022b; Chang et al. 2022; Deng et al. 2022; Liu et al. 2023). However, despite their expressive power, transformers may still face challenges in capturing local image structures or handling high-resolution inputs efficiently due to architectural and computational constraints.

The rise of diffusion models, particularly Stable Diffu-

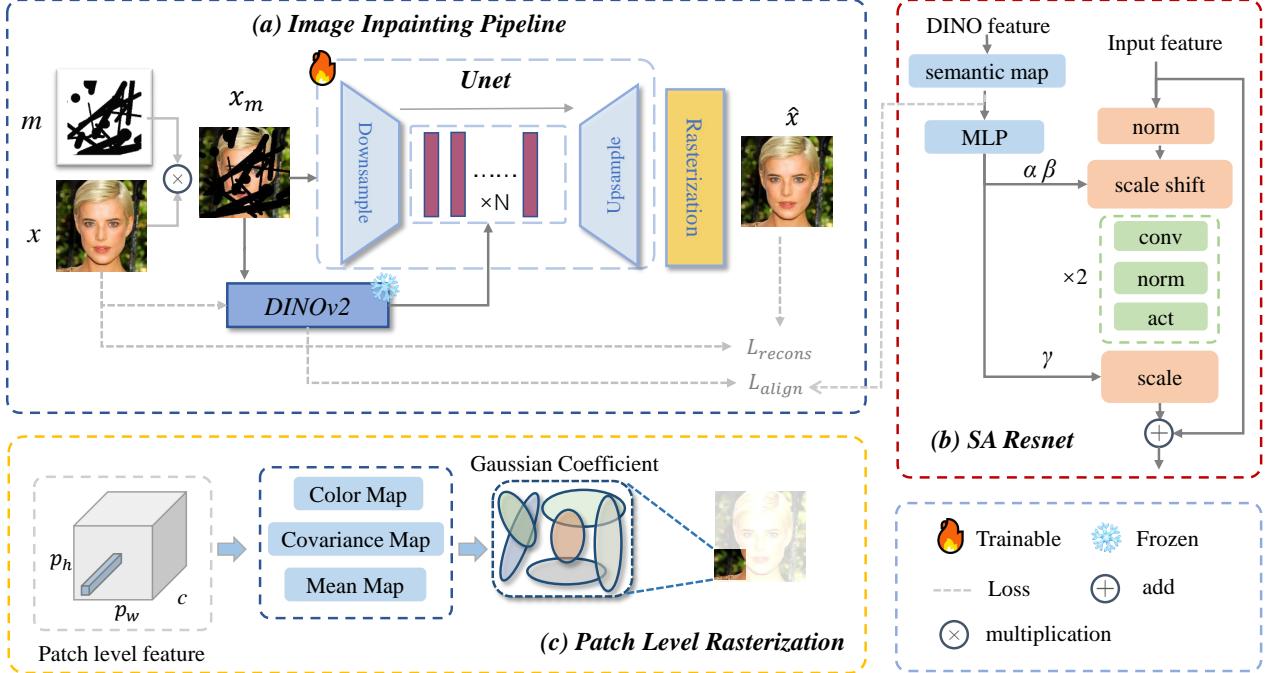


Figure 2: Overview of our proposed framework: (a) The overall pipeline consists of a U-Net architecture, DINO-based semantic alignment, and a differentiable rasterization module. (b) The Semantic Alignment (SA) ResNet integrates high-level semantic priors from DINO using an AdaLN-based modulation. (c) The Patch-Level Rasterization module transforms patch-wise Gaussian parameters into continuous representations, which are then composited to generate the complete image.

sion (Rombach et al. 2022), has enabled strong generative priors for inpainting. Methods (Xie et al. 2023; Ju et al. 2024; Deng et al. 2025) project corrupted images into latent spaces for restoration, while others (Yu et al. 2023; Wasserman et al. 2024; Tianyidan et al. 2025) incorporate LLMs for controllable, semantically guided inpainting. Despite their quality, these models often suffer from slow inference and high computational costs.

Recent attempts to integrate Gaussian Splatting with diffusion models (Fein-Ashley and Fein-Ashley 2024) show limited performance on small, outdated datasets. In contrast, our work pioneers a dedicated 2D Gaussian Splatting framework for inpainting, offering a lightweight, semantically aware framework.

3 Method

The image inpainting task aims to reconstruct a complete image from a partial image with reasonable content, requiring overall coherence and clarity in the reconstructed result. In this section, we present our end-to-end encoder–rasterization framework with semantic alignment for image inpainting. We first review the basis of Gaussian splatting, then describe our complete inpainting pipeline with an efficient patch-level rasterization strategy with overlap smoothing. Finally, we present our semantic alignment strategy, which leverages high-level priors to guide feature adaptation and improve global consistency.

3.1 Preliminary: Gaussian Splatting

Gaussian Splatting (GS) (Kerbl et al. 2023) has demonstrated remarkable capabilities in the field of 3D view synthesis and is naturally suited for visual representation tasks due to its hybrid discrete-continuous nature. The image is represented by numerous 2D Gaussians, each Gaussian characterized by mean $\mu_i \in \mathbb{R}^2$, covariance matrix $\Sigma_i \in \mathbb{R}^{2 \times 2}$, coefficient $\sigma_i \in \mathbb{R}^1$, and color $c_i \in \mathbb{R}^3$.

Typically, the covariance matrix of a Gaussian should be positive semi-definite. Therefore, inspired by (Zhang et al. 2024), Σ_i can be factorized into the product of a lower triangular matrix $L_i \in \mathbb{R}^{2 \times 2}$ and its conjugate transpose L_i^T , which ensures positive semi-definiteness and reduces the covariance representation from 4 to 3. We denote this parameter set as the Gaussian space Θ .

$$\Sigma_i = L_i L_i^T \quad \Theta = \{\mu, L, c, \tilde{\sigma}\} \quad (1)$$

Primarily due to the favorable mathematical properties of the Gaussian Mixture Model(GMM) (Reynolds et al. 2009), complex distributions can be constructed by multiple Gaussian kernels. Given the Gaussian field at position (x, y) , we represent the pixel coordinate as a 2D vector $\mathbf{p} = [x, y]^T$. The pixel value is computed as the sum of Gaussians:

$$I_{\mathbf{p}} = \sum_i c_i \sigma_i \exp \left(-\frac{1}{2} (\mathbf{p} - \mu_i)^T \Sigma_i^{-1} (\mathbf{p} - \mu_i) \right) \quad (2)$$

Therefore, given enough Gaussians, the image can be generated in the above way, where the relevant parameters of the Gaussians can generally be trainable.

3.2 2D Gaussian Splatting Framework

Building upon the above formulations, our pipeline consists of two main stages: (1) Gaussian feature encoding, where the input image is projected into a Gaussian parameter space; and (2) Rasterization-based rendering, which reconstructs the complete image by rendering the learned Gaussian fields. A conceptual overview is illustrated in Figure 2(a).

Gaussian feature encoding Traditional methods often rely on latent feature embeddings obtained from neural networks, followed by a trainable decoder that maps these features back to pixel space. However, this approach typically relies on the training performance of the decoder and struggles to explicitly implement pixel-level continuity naturally. Therefore, our method models the image feature as the 2D Gaussian features, which are concrete in Gaussian space and continuous in each Gaussian domain. Each Gaussian kernel defines a smooth and differentiable field over the image plane, allowing the model to naturally propagate visual information from observed to missing regions. This spatially overlapping and continuous formulation is fundamentally advantageous for achieving pixel-level continuity.

Formally, given a masked input image $I_{\text{mask}} \in \mathbb{R}^{3 \times H \times W}$, we employ a Unet encoder to extract a set of N Gaussian features. The encoder outputs a dense feature map $F_g \in \mathbb{R}^{C' \times H \times W}$ which is then downsampled via a strided convolution layer to a Gaussian-level feature $F'_g \in \mathbb{R}^{N \times C'}$, where N denotes the number of Gaussian kernels.

The encoder’s hierarchical structure, combined with skip connection, captures rich context while enhancing training stability. Moreover, the Gaussian mean μ , which has been empirically observed to be sensitive in the training process, is initialized uniformly across the 2D plane as μ_{fix} and only the mean offset μ_{bias} is learned.

Parameter Decoding Instead of a neural network decoder outputting discrete pixel values, we decode Gaussian parameters from F'_g using a set of lightweight multilayer perceptrons (MLPs):

$$\mu_{\text{bias}} = E_\mu(F'_g), \quad c = E_c(F'_g), \quad l = E_l(F'_g) \quad (3)$$

where $\mu_{\text{bias}} \in \mathbb{R}^{N \times 2}$ denotes the learnable positional offsets, $c \in \mathbb{R}^{N \times 3}$ and $l \in \mathbb{R}^{N \times 3}$ represent the color and covariance parameters. To constrain the spatial extent of μ_{bias} , we apply a \tanh activation, restricting its values to the range $[-1, 1]$. The final positions are obtained by adding these learnable offsets to uniformly initialized positions: $\mu = \mu_{\text{bias}} + \mu_{\text{fix}}$.

These parameters are subsequently rasterized onto the image plane as described in Section 3.1, and their soft overlapping nature guarantees smooth color and intensity transitions across pixels, ensuring seamless image inpainting.

3.3 Patch-level Rasterization

However, achieving high-fidelity image representation with Gaussian kernels typically requires a large number of Gaussians, which scales with image resolution and leads to significant GPU memory consumption during rasterization. Moreover, directly generating Gaussian parameters from high-dimensional latent features for the entire image results in

a large parameter set, further exacerbating memory pressure and slowing down rendering.

To mitigate this, we leverage the spatial locality of natural images and propose a patch-level 2D rasterization approach. Instead of managing a single global Gaussian set, we divide the image into multiple non-overlapping patches and assign a dedicated set of Gaussians to each.

Formally, an image $I \in \mathbb{R}^{H \times W}$ is conceptually divided into a grid of $N_p = \frac{H}{p} \times \frac{W}{p}$ non-overlapping patches, each of nominal size $p \times p$. For each patch (i, j) corresponding to grid coordinates, we maintain a dedicated set of N_{patch} Gaussian kernels. Then, parameter space Θ transforms into patch level space:

$$\mu \in \mathbb{R}^{N_p \times N \times 2}, \quad c \in \mathbb{R}^{N_p \times N \times 3}, \quad l \in \mathbb{R}^{N_p \times N \times 3} \quad (4)$$

While it might not reduce the total number of Gaussians ($N_p \times N_{\text{patch}}$) required for equivalent quality compared to a global approach, it reduces the number of Gaussians that need to be loaded and processed simultaneously during the rasterization of any single patch. This reduction in concurrent memory demand alleviates GPU memory pressure. Furthermore, since each patch can be rendered independently, the rasterization process is highly parallelizable across patches, leading to faster overall rendering times.

An inherent challenge with processing images in independent patches is the visible discontinuities at the patch boundaries, as the representation of one patch is independent of its neighbors. To ensure smoothness across patch borders, we employ a patch overlap strategy: during rendering, each patch is processed not just over its nominal $p \times p$ area, but over an extended region that includes a border of a pixels on all four sides. This means each patch (i, j) is rasterized over an area of size $(p + 2a) \times (p + 2a)$, using its parameter set $\Theta_{i,j}$, to produce a rendered patch $R_{i,j}$. The rendered patches $R_{i,j}$ thus overlap with their neighbors. To construct the final image, we retain the central $(p - 2a) \times (p - 2a)$ region of each $R_{i,j}$ and blend the overlapping border areas with neighboring patches. Blending is typically done using weighted averages based on pixel distance from the patch center or boundary, ensuring smooth transitions.

3.4 Feature Adaptation For Semantic Guidance

We hypothesize that semantic priors from pretrained models can provide valuable global context for image inpainting, enabling more cohesive and semantically faithful completion. To this end, we adopt features extracted from the pretrained DINOv2 model (Oquab et al. 2023) as guidance signals within our framework.

A critical challenge arises from the fact that the input to the model is a masked image, and it is unclear whether features extracted from such incomplete inputs are meaningful or reliable. To investigate this, we conduct experiments illustrated in Figure 3, which shows that DINO features extracted from lightly masked images still retain rich semantic information and can effectively guide inpainting. However, as the mask size increases, the extracted features become increasingly corrupted and less informative. This degradation limits their effectiveness in directly guiding the inpainting process for large missing regions.

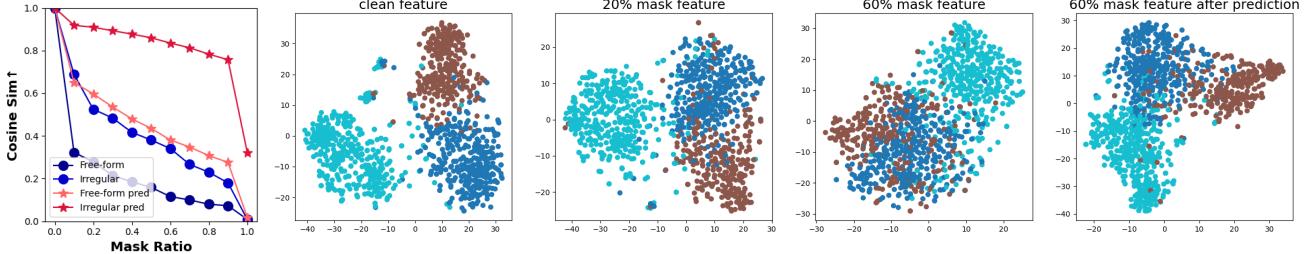


Figure 3: Representation Gap with Mask Image: (a) Cosine similarity between masked and unmasked DINOv2 features drops with higher mask ratios, while the mapping block preserves similarity. (b) t-SNE plots show improved cluster separation and compactness with the block, indicating enhanced semantic consistency and reduced representation degradation under masking.

To address this, we propose a simple yet effective feature adaptation module that transforms noisy masked features into semantically coherent representations. Specifically, we employ a lightweight MLP to learn a mapping from the masked feature space to an estimated clean feature space. As demonstrated in Figure 3, this transformation significantly improves feature quality and enables robust semantic guidance across a range of mask sizes. The adapted features thus serve as conditional inputs to the inpainting network.

For integrating the adapted semantic features into the inpainting process, we adopt the Adaptive Layer Normalization (AdaLN) mechanism (Karras, Laine, and Aila 2019), which is both parameter-efficient and capable of modulating network activations globally.

As shown in Figure 2, given a predicted semantic feature vector $f_{\text{pred}} \in \mathbb{R}^{C_0}$ and a hidden feature map $f \in \mathbb{R}^{C \times H \times W}$, the AdaLN operation is defined as:

$$\text{AdaLN}(f_{\text{pred}}, f) = B(LN(x) \times \alpha_{\text{pred}} + \beta_{\text{pred}}) \times \gamma_{\text{pred}} \quad (5)$$

where $B(\cdot)$ denotes the main processing block, and $\alpha_{\text{pred}}, \beta_{\text{pred}}, \gamma_{\text{pred}} \in \mathbb{R}^d$ are learned affine parameters obtained via linear projection from f_{pred} .

To explicitly align the predicted features with the ground truth semantic representations, we use a feature alignment loss based on negative cosine similarity:

$$\mathcal{L}_{\text{align}} = -\frac{f_{\text{clean}} \cdot f_{\text{pred}}}{|f_{\text{clean}}| \cdot |f_{\text{pred}}|} \quad (6)$$

which encourages the predicted features to align with the direction of the clean features, thereby enhancing global semantic consistency during inpainting.

4 Experiments

4.1 Implementation Detail

Architecture We adopt a simple convolutional U-Net as the image encoder, consisting of 3 downsampling layers, 9 bottleneck (middle) layers, and 3 upsampling layers. For semantic feature alignment, we utilize ViT-14 pretrained weights from DINOv2 and inject the extracted features into all 9 bottleneck layers of the encoder. To enhance training stability and accelerate convergence, especially during the early stages, we incorporate the Gaussian prior initialization strategy introduced in (Peng et al. 2025). The network takes

as input only the masked images, without explicitly providing the binary mask itself.

Loss Function The model is optimized using a composite loss function that balances reconstruction accuracy, perceptual quality, adversarial realism, and semantic alignment. The total objective is defined as:

$$\mathcal{L}_{\text{total}} = w_1 \mathcal{L}_{\text{recons}} + w_2 \mathcal{L}_{\text{gan}} + w_3 \mathcal{L}_{\text{lpipl}} + w_4 \mathcal{L}_{\text{align}} \quad (7)$$

where the weights are empirically set as $w_1 : w_2 : w_3 : w_4 = 1 : 0.3 : 3 : 1$. The $\mathcal{L}_{\text{recons}}$ loss measures the absolute difference between the rendered pyramid and the ground truth. The \mathcal{L}_{gan} is an adversarial loss employing a discriminator, similar to those used in SD (Rombach et al. 2022), trained to distinguish between images rendered from the predicted Gaussians and real images. The $\mathcal{L}_{\text{lpipl}}$ loss captures perceptual similarity based on deep features. The $\mathcal{L}_{\text{align}}$ enforces semantic consistency by minimizing the discrepancy between the DINO features of the masked input and those of the clean image.

Datasets and Metrics We conduct experiments on the two most commonly used datasets: Celeba-HQ (Karras et al. 2017) and Places2 (Zhou et al. 2017). Celeba-HQ contains 30,000 high-quality face images. We use 28,000 images for training and 2,000 images for evaluation. The train-test split follows LaMa (Suvorov et al. 2022). We choose Places365, which has 1.8M natural images for training and commonly used 36,500 validation images for evaluation. To test the inpainting results, we choose the following metrics: FID (Heusel et al. 2017), LPIPS (Zhang et al. 2018).

Training Details All models are trained with a batch size of 64 using the Adam optimizer and an initial learning rate of 2×10^{-4} . The default patch size is set to 16×16 , and each Gaussian element is represented with a 12-dimensional hidden embedding. All experiments are conducted on 8 NVIDIA A800 GPUs.

4.2 Qualitative Comparison

Figure 4 presents qualitative comparisons across various image inpainting methods on two datasets: CelebA-HQ (top two rows) and Places2 (bottom two rows). It is evident that some methods, such as Latent-Code, RePaint, and Pluralistic, exhibit noticeable artifacts or semantic inconsistencies,

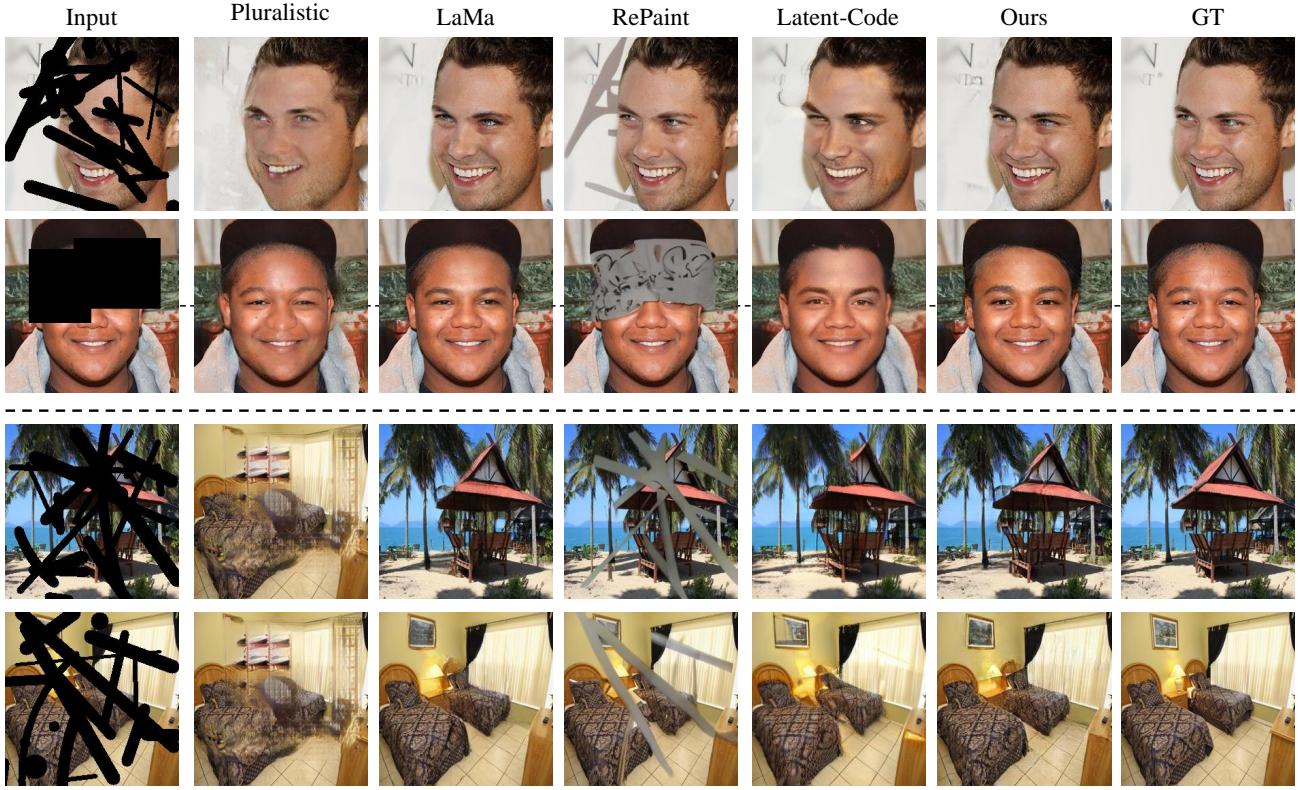


Figure 4: Qualitative Results. The top two rows show face inpainting results from the CelebA-HQ dataset, while the bottom two rows display natural scenes from the Places2 dataset with either irregular or regular patterns.

particularly in complex textures and facial features. In contrast, our method produces visually coherent and semantically plausible completions.

4.3 Quantitative Comparison

To evaluate our method, we compare against representative baselines: Latent-Code (Chen and Zhao 2024), Pluralistic (Zheng, Cham, and Cai 2019), ZITS++ (Cao, Dong, and Fu 2023), RePaint (Lugmayr et al. 2022), LaMa (Suvorov et al. 2022), MAT (Li et al. 2022b), as summarized in Table 1. Experiments are conducted on both regular and irregular masks of varying sizes, using publicly available checkpoints and identical image-mask pairs for fair comparison.

Our method matches or outperforms state-of-the-art baselines in both fidelity and perceptual quality. It preserves identity well on face datasets and generalizes effectively to complex natural scenes. While our FID is slightly higher than LaMa’s, this may result from its frequency-domain modeling and multi-scale fusion, which better capture global structure. Our approach, by contrast, focuses on local continuity and semantic coherence. As FID emphasizes distribution alignment, it may slightly favor methods like LaMa.

4.4 Ablation Study

This study aims to investigate the impact of our key components through a series of ablation experiments. We adopt ImageNet-100 (Deng et al. 2009), a subset of ImageNet that focuses on natural scenes, due to its balance of efficiency and content diversity. The dataset comprises 130,000 images for

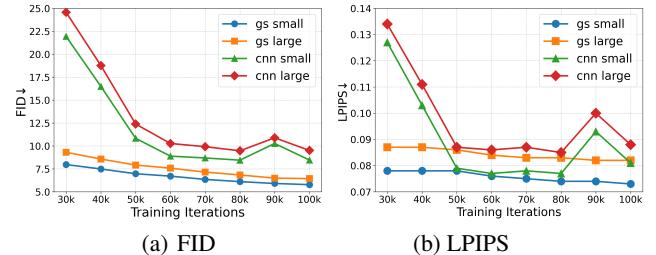


Figure 5: Convergence Speed comparison between CNN decoder and Gaussian decoder

training and 5,000 images for testing. For fair comparison, all models are trained for 100,000 steps and evaluated on the same set of image-mask pairs.

DINO Feature To assess the effectiveness of the feature alignment module, firstly, we remove the semantic adaptation component and directly use the raw features from DINO. This results in a noticeable degradation in image quality, particularly under large-mask conditions, indicating that unadapted features are insufficient for guiding the inpainting process effectively. Next, we remove the entire semantic alignment module, which leads to a further decline in performance, with more artifacts and structural inconsistencies observed in the inpainted regions as shown in Figure 5.

CNN Decoder To assess the rasterization-based decoder, we replace it with a CNN decoder using transposed con-

Method	Celeba-HQ						Places2			
	Small		Large		Regular		Small		Large	
	FID↓	LPIPS↓								
Latent-Code	24.04	0.098	26.39	0.120	27.48	0.125	3.21	0.097	5.31	0.131
Pluralistic	16.59	0.198	17.09	0.201	16.98	0.203	9.83	0.197	16.26	0.226
ZITS++	-	-	-	-	-	-	3.87	0.118	6.13	0.376
RePaint	7.26	0.066	10.06	0.071	9.71	0.069	11.16	0.102	20.62	0.117
LaMa	5.26	0.037	8.91	0.057	4.86	0.053	1.43	0.061	3.60	0.104
MAT	9.83	0.064	9.84	0.065	7.15	0.068	1.51	0.067	3.47	0.087
Ours	6.38	0.028	9.45	0.054	7.03	0.043	2.61	0.056	5.03	0.094

Table 1: Quantitative evaluation on Places2 and CelebA-HQ datasets. The best and second best results are in red and blue.

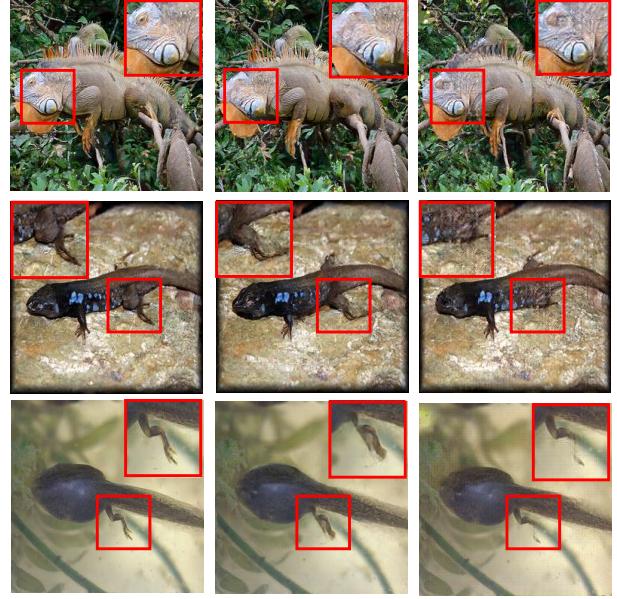
Settings	Small		Large	
	FID↓	LPIPS↓	FID↓	LPIPS↓
Full Model	5.78	0.073	6.44	0.082
w/o dino map	7.34	0.076	8.37	0.085
w/o dino	8.73	0.077	9.92	0.086
CNN decoder	8.46	0.081	9.51	0.088
GS-100	7.22	0.077	8.36	0.085
GS-196	6.38	0.075	7.16	0.084
SA-Adaln-no- γ	7.28	0.079	8.04	0.087
SA-Concat	7.27	0.082	8.40	0.082

Table 2: Ablation study results on ImageNet100 dataset.

volutions on the same patch-level features. Both decoders have comparable parameter counts to ensure fairness. As shown in Table 2, the CNN decoder suffers a notable performance drop. Figure 6 reveals visual artifacts, while Figure 5 shows slower convergence and training instability, especially in early stages. In contrast, our Gaussian decoder enables faster, more stable learning with superior results.

Gaussian Numbers The number of Gaussians is a critical hyperparameter that significantly affects generation quality. To study its impact, we train models with 100 and 196 Gaussians and compare them against our baseline setting of 324 Gaussians, as shown in Table 2. As expected, reducing the number of Gaussians leads to a corresponding decline in performance. Intuitively, increasing the number of Gaussians could significantly improve quality by offering finer spatial resolution. However, this significantly raises computational cost and model size, making it impractical for efficient training and inference. We leave this for future work.

Condition Methods To assess the effectiveness of our AdaLN module, we conduct two ablation experiments. First, we remove the scaling parameter γ , resulting in severe training instability. As shown in Table 2, the reported results are from step 50,000, beyond which training completely collapses. Notably, even before the breakdown, the model underperforms the baseline across several metrics. In the second experiment, we replace AdaLN with a simple concatenation-based fusion while keeping the number of parameters approximately the same to ensure a fair comparison. This variant yields a noticeable drop in performance across all evaluation metrics, further demonstrating the superiority of our AdaLN design.



(a) full model (b) w/o dino (c) cnn decoder

Figure 6: Ablation study. Visual comparison of inpainting results under different configurations: (a) the full model (b) removing DINO-based semantic guidance (c) replacing our rasterization-based decoder with a CNN-based decoder.

5 Conclusion

In this paper, we present a novel patch-level 2D Gaussian Splatting (2DGS) framework with semantic alignment for image inpainting, effectively addressing the challenge of coherent completion in missing regions. Our approach encodes incomplete images into 2D Gaussian parameters using a lightweight CNN-based U-Net and reconstructs them through a learnable rasterization pipeline. The key contributions include a semantic feature alignment strategy that leverages DINO-based priors to guide the inpainting process, and a patch-level rasterization mechanism that significantly reduces GPU memory usage and improves inference efficiency by operating on localized image blocks. Extensive experiments demonstrate that our method achieves competitive performance, while ablation studies validate the effectiveness of each key component. This work highlights the strong potential of efficient, Gaussian-based representations for realistic image restoration and broader visual synthesis.

References

- Cao, C.; Dong, Q.; and Fu, Y. 2023. Zits++: Image inpainting by improving the incremental transformer on structural priors. *IEEE transactions on pattern analysis and machine intelligence*, 45(10): 12667–12684.
- Cao, J.; Wang, Q.; Xian, Y.; Li, Y.; Ni, B.; Pi, Z.; Zhang, K.; Zhang, Y.; Timofte, R.; and Van Gool, L. 2023. Ciaosr: Continuous implicit attention-in-attention network for arbitrary-scale image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1796–1807.
- Chang, H.; Zhang, H.; Jiang, L.; Liu, C.; and Freeman, W. T. 2022. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11315–11325.
- Chen, D.; Chen, L.; Zhang, Z.; and Zhang, L. 2025. Generalized and Efficient 2D Gaussian Splatting for Arbitrary-scale Super-Resolution. *arXiv preprint arXiv:2501.06838*.
- Chen, H.; and Zhao, Y. 2024. Don’t Look into the Dark: Latent Codes for Pluralistic Image Inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7591–7600.
- Chen, Y.; Liu, S.; and Wang, X. 2021. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8628–8638.
- Chen, Z.; Wang, F.; Wang, Y.; and Liu, H. 2024. Text-to-3d using gaussian splatting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 21401–21412.
- Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.
- Deng, J.; Wu, X.; Yang, Y.; Zhu, C.; Wang, S.; and Wu, Z. 2025. Acquire and then Adapt: Squeezing out Text-to-Image Model for Image Restoration. *arXiv preprint arXiv:2504.15159*.
- Deng, Y.; Hui, S.; Zhou, S.; Meng, D.; and Wang, J. 2022. T-former: An efficient transformer for image inpainting. In *Proceedings of the 30th ACM international conference on multimedia*, 6559–6568.
- Dong, J.; Wang, C.; Zheng, W.; Chen, L.; Lu, J.; and Tang, Y. 2025. GaussianToken: An Effective Image Tokenizer with 2D Gaussian Splatting. *arXiv preprint arXiv:2501.15619*.
- Efros, A. A.; and Leung, T. K. 1999. Texture synthesis by non-parametric sampling. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, 1033–1038. IEEE.
- Esser, P.; Rombach, R.; and Ommer, B. 2021. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12873–12883.
- Fein-Ashley, J.; and Fein-Ashley, B. 2024. Diffusion Models with Anisotropic Gaussian Splatting for Image Inpainting. *arXiv:2412.01682*.
- Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.
- Guo, X.; Yang, H.; and Huang, D. 2021. Image inpainting via conditional texture and structure dual generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 14134–14143.
- He, K.; and Sun, J. 2014. Image completion approaches using the statistics of similar patches. *IEEE transactions on pattern analysis and machine intelligence*, 36(12): 2423–2435.
- Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.
- Hu, J.; Xia, B.; Chen, B.; Yang, W.; and Zhang, L. 2025. Gaussiansr: High fidelity 2d gaussian splatting for arbitrary-scale image super-resolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 3554–3562.
- Huang, Y.-H.; Sun, Y.-T.; Yang, Z.; Lyu, X.; Cao, Y.-P.; and Qi, X. 2024. Sc-gs: Sparse-controlled gaussian splatting for editable dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4220–4230.
- Jiang, C.; Jia, H.; Xu, H.; Ye, W.; Dong, M.; Yan, M.; Zhang, J.; Huang, F.; and Zhang, S. 2024. Maven: An effective multi-granularity hybrid visual encoding framework for multimodal large language model. *Advances in Neural Information Processing Systems*, 37: 101992–102010.
- Jiao, Y.; Qiu, H.; Jie, Z.; Chen, S.; Chen, J.; Ma, L.; and Jiang, Y.-G. 2025. Unitoken: Harmonizing multimodal understanding and generation through unified visual encoding. *arXiv preprint arXiv:2504.04423*.
- Johnson, J.; Alahi, A.; and Fei-Fei, L. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, 694–711. Springer.
- Ju, X.; Liu, X.; Wang, X.; Bian, Y.; Shan, Y.; and Xu, Q. 2024. Brushnet: A plug-and-play image inpainting model with decomposed dual-branch diffusion. In *European Conference on Computer Vision*, 150–168. Springer.
- Karras, T.; Aila, T.; Laine, S.; and Lehtinen, J. 2017. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.
- Karras, T.; Laine, S.; and Aila, T. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4401–4410.
- Keetha, N.; Karhade, J.; Jatavallabhula, K. M.; Yang, G.; Scherer, S.; Ramanan, D.; and Luiten, J. 2024. Splatam: Splat track & map 3d gaussians for dense rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21357–21366.

- Kerbl, B.; Kopanas, G.; Leimkühler, T.; and Drettakis, G. 2023. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4): 139–1.
- Li, H.; Dai, T.; Li, Y.; Zou, X.; and Xia, S.-T. 2022a. Adaptive local implicit image function for arbitrary-scale super-resolution. In *2022 IEEE International Conference on Image Processing (ICIP)*, 4033–4037. IEEE.
- Li, W.; Lin, Z.; Zhou, K.; Qi, L.; Wang, Y.; and Jia, J. 2022b. Mat: Mask-aware transformer for large hole image inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10758–10768.
- Liu, G.; Reda, F. A.; Shih, K. J.; Wang, T.-C.; Tao, A.; and Catanzaro, B. 2018. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European conference on computer vision (ECCV)*, 85–100.
- Liu, W.; Cun, X.; Pun, C.-M.; Xia, M.; Zhang, Y.; and Wang, J. 2023. Coordfill: Efficient high-resolution image inpainting via parameterized coordinate querying. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 1746–1754.
- Lugmayr, A.; Danelljan, M.; Romero, A.; Yu, F.; Timofte, R.; and Van Gool, L. 2022. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11461–11471.
- Luiten, J.; Kopanas, G.; Leibe, B.; and Ramanan, D. 2024. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *2024 International Conference on 3D Vision (3DV)*, 800–809. IEEE.
- Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106.
- Oquab, M.; Dariset, T.; Moutakanni, T.; Vo, H. V.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; Howes, R.; Huang, P.-Y.; Xu, H.; Sharma, V.; Li, S.-W.; Galuba, W.; Rabbat, M.; Assran, M.; Ballas, N.; Synnaeve, G.; Misra, I.; Jegou, H.; Mairal, J.; Labatut, P.; Joulin, A.; and Bojanowski, P. 2023. DINov2: Learning Robust Visual Features without Supervision.
- Pathak, D.; Krahenbuhl, P.; Donahue, J.; Darrell, T.; and Efros, A. A. 2016. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2536–2544.
- Peng, L.; Wu, A.; Li, W.; Xia, P.; Dai, X.; Zhang, X.; Di, X.; Sun, H.; Pei, R.; Wang, Y.; et al. 2025. Pixel to gaussian: Ultra-fast continuous super-resolution with 2d gaussian modeling. *arXiv preprint arXiv:2503.06617*.
- Reynolds, D. A.; et al. 2009. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663): 3.
- Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.
- Suvorov, R.; Logacheva, E.; Mashikhin, A.; Remizova, A.; Ashukha, A.; Silvestrov, A.; Kong, N.; Goka, H.; Park, K.; and Lempitsky, V. 2022. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2149–2159.
- Tian, C.; Tao, C.; Dai, J.; Li, H.; Li, Z.; Lu, L.; Wang, X.; Li, H.; Huang, G.; and Zhu, X. 2023. Addp: Learning general representations for image recognition and generation with alternating denoising diffusion process. *arXiv preprint arXiv:2306.05423*.
- Tianyidan, X.; Ma, R.; Wang, Q.; Ye, X.; Liu, F.; Tai, Y.; Zhang, Z.; Wang, L.; and Yi, Z. 2025. Anywhere: A Multi-Agent Framework for User-Guided, Reliable, and Diverse Foreground-Conditioned Image Generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 7410–7418.
- Wasserman, N.; Rotstein, N.; Ganz, R.; and Kimmel, R. 2024. Paint by inpaint: Learning to add image objects by removing them first. *arXiv preprint arXiv:2404.18212*.
- Wu, H.; Ni, N.; and Zhang, L. 2023. Learning dynamic scale awareness and global implicit functions for continuous-scale super-resolution of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 61: 1–15.
- Xie, S.; Zhang, Z.; Lin, Z.; Hinz, T.; and Zhang, K. 2023. Smartbrush: Text and shape guided object inpainting with diffusion model. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 22428–22437.
- Yu, J.; Lin, Z.; Yang, J.; Shen, X.; Lu, X.; and Huang, T. S. 2018. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5505–5514.
- Yu, J.; Lin, Z.; Yang, J.; Shen, X.; Lu, X.; and Huang, T. S. 2019. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF international conference on computer vision*, 4471–4480.
- Yu, T.; Feng, R.; Feng, R.; Liu, J.; Jin, X.; Zeng, W.; and Chen, Z. 2023. Inpaint anything: Segment anything meets image inpainting. *arXiv preprint arXiv:2304.06790*.
- Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 586–595.
- Zhang, X.; Ge, X.; Xu, T.; He, D.; Wang, Y.; Qin, H.; Lu, G.; Geng, J.; and Zhang, J. 2024. Gaussianimage: 1000 fps image representation and compression by 2d gaussian splatting. In *European Conference on Computer Vision*, 327–345. Springer.
- Zheng, C.; Cham, T.-J.; and Cai, J. 2019. Pluralistic image completion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1438–1447.
- Zhou, B.; Lapedriza, A.; Khosla, A.; Oliva, A.; and Torralba, A. 2017. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6): 1452–1464.

A Implementation Details

A.1 Hyperparameter

We provide the experiment details on CelebA-HQ and Places2 datasets for reproducibility below:

Settings	CelebA-HQ	Places2
Image size	256	256
Patch size	16×16	16×16
Gaussian per patch	324	324
Hidden dimension per gs	12	12
Overlap pad	1	1
Batch size	64	64
Learning rate	2e-4	2e-4
Training steps	60k	200k
Optimizer	Adam	Adam
GAN loss	hinge	hinge
D learning rate	2e-4	2e-4
D optimizer	Adam	Adam
Reconstruction loss weight	1	1
Perceptual loss weight	2	3
GAN loss weight	0.2	0.3
Alignment loss weight	1	1

Table 3: Hyperparameters

We refer to the small mask as $20\% \sim 40\%$ ratio and the large mask as $40\% \sim 60\%$ ratio.

A.2 Mask Ratio Strategy

When training the model with varying mask ratios, we progressively increase the masking ratio to help the model adapt more easily. After a certain number of iterations, we begin randomly sampling the ratio from the full range to prevent the model from underperforming on smaller ratios. Concretely, for the CelebA-HQ dataset, we increase the masking ratio every 20 epochs, while for the Places2 dataset, we do so every epoch due to its larger scale.

B Social impacts

B.1 Positive impacts

- Advancements in AI: A New paradigm enhances the development of a data tokenizer capable of assisting in various tasks.
- Practicality: The method can provide assistance for the image restoration field, aiding in future restoration work and helping people repair damaged data.

B.2 Negative impacts

- Job Displacement: Advanced AI could potentially displace jobs in the data restoration field, necessitating consideration of economic and societal impacts.
- Misuse: Misuse of our model may lead to data fraud, information infringement, incorrect guidance on the internet, resulting in economic and ethical issues.

C Discussion

While our approach achieves strong results in both semantic consistency and visual fidelity, it currently lacks the explicit controllability. These methods often benefit from multi-modal inputs, such as textual prompts or structural cues, that enable more flexible and user-guided generation. In contrast, our design operates without external guidance, which limits its applicability in scenarios requiring fine-grained semantic control or interactive editing. Enhancing our framework with cross-modal conditioning mechanisms is a compelling direction for future exploration.

D More Qualitative Results

To further validate the robustness and generalization of our method, we present qualitative results across diverse datasets and mask settings. As shown in Figures 8–11, our approach consistently produces semantically coherent and visually realistic completions.

On CelebA-HQ (Figure 8), our model effectively preserves facial structure and identity, even under large, irregular masks. For complex natural scenes in Places2 (Figure 9), it reconstructs rich textures and spatial layouts. Results on FFHQ (Figure 10) further highlight its ability to handle high-resolution portraits with fine details. Finally, the performance on the challenging and diverse ImageNet-100 dataset (Figure 11) demonstrates strong generalization to object-centric images with varied semantics.

Overall, these results confirm that our method not only maintains semantic fidelity across domains, but also adapts well to arbitrary mask distributions—underscoring its potential for real-world inpainting applications.

E Real-world Scenario Inpainting

As illustrated in Figure 7, we manually remove specific foreground objects from natural images using segmentation masks to simulate realistic inpainting scenarios. This setup mirrors common image editing tasks such as object removal. The masks are designed to selectively occlude frequently encountered elements in outdoor scenes, such as tents, chairs, signage, and people, thereby creating a practical testbed that closely reflects real-world use cases. The second column shows the masked inputs fed into the model, while the final column presents the corresponding inpainted outputs. This visualization not only showcases our model’s ability to reconstruct complex backgrounds with high visual fidelity, but also highlights its effectiveness in preserving structural continuity and semantic coherence in the absence of salient foreground content. Furthermore, the diversity in object shapes and scene compositions underscores the robustness and applicability of our method across a wide range of real-world scenarios.

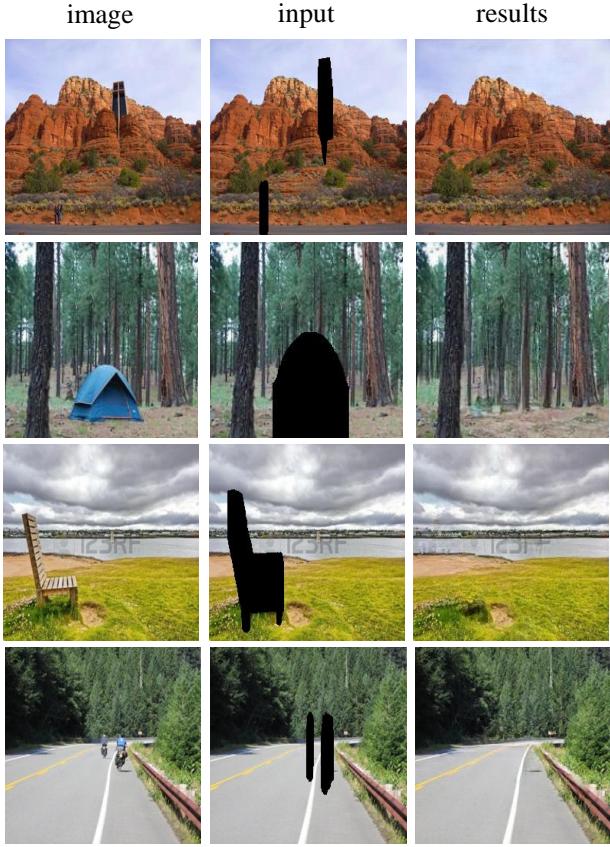


Figure 7: More results on Real-world Scenarios

F Efficiency

To assess the computational efficiency of our method, we compare its inference speed with several representative inpainting approaches, as summarized in Table 4. All methods are evaluated under identical hardware settings to ensure a fair comparison. Our approach generates significantly faster than diffusion-based or transformer-heavy methods,

Method	Speed(ms)
Ours	32.52
LaMa	15.80
MAT	65.75
Latent-Code	45.67
RePaint	79035.84

Table 4: Average inference time per image.

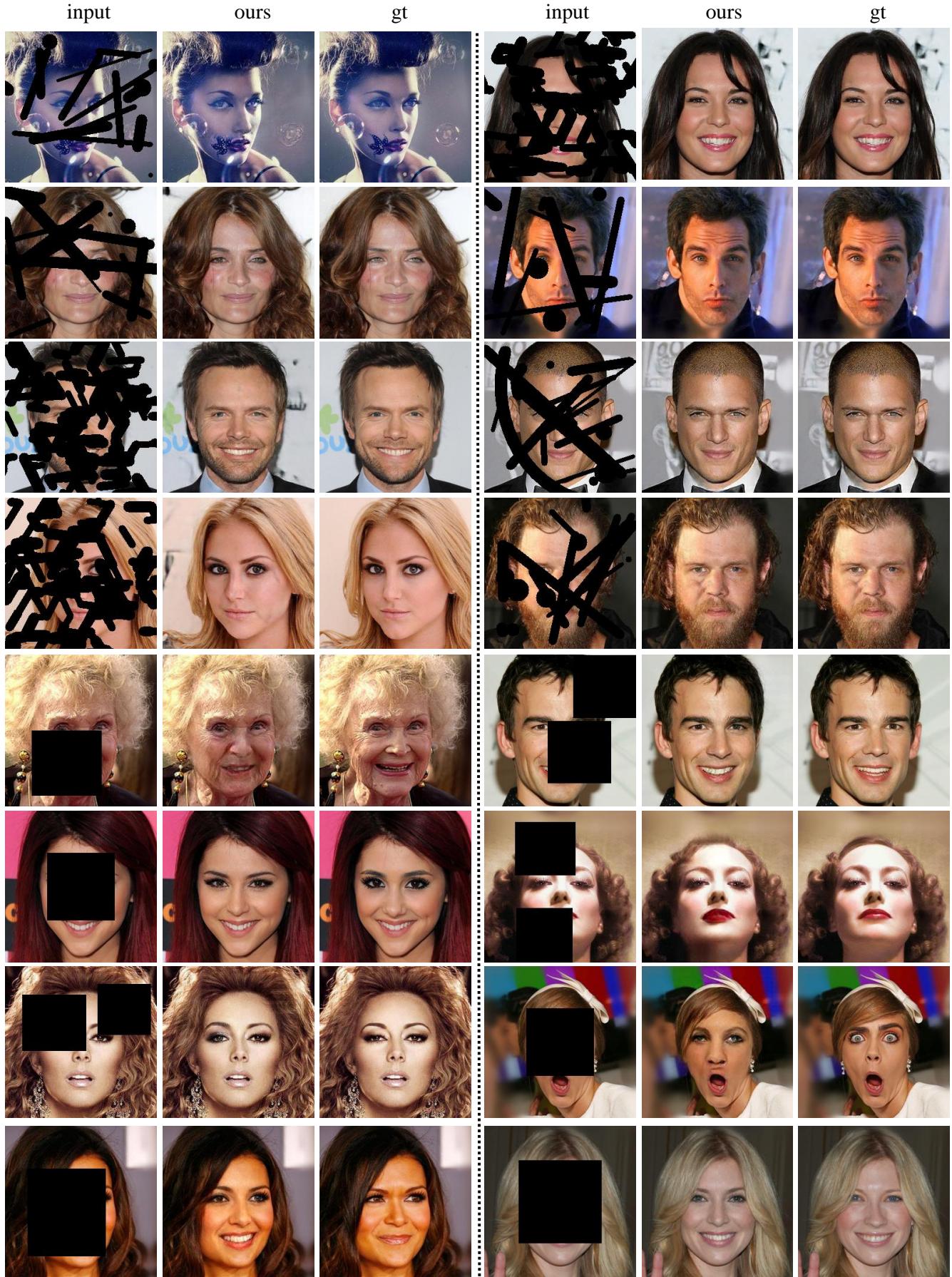


Figure 8: Qualitative results on CelebA-HQ.

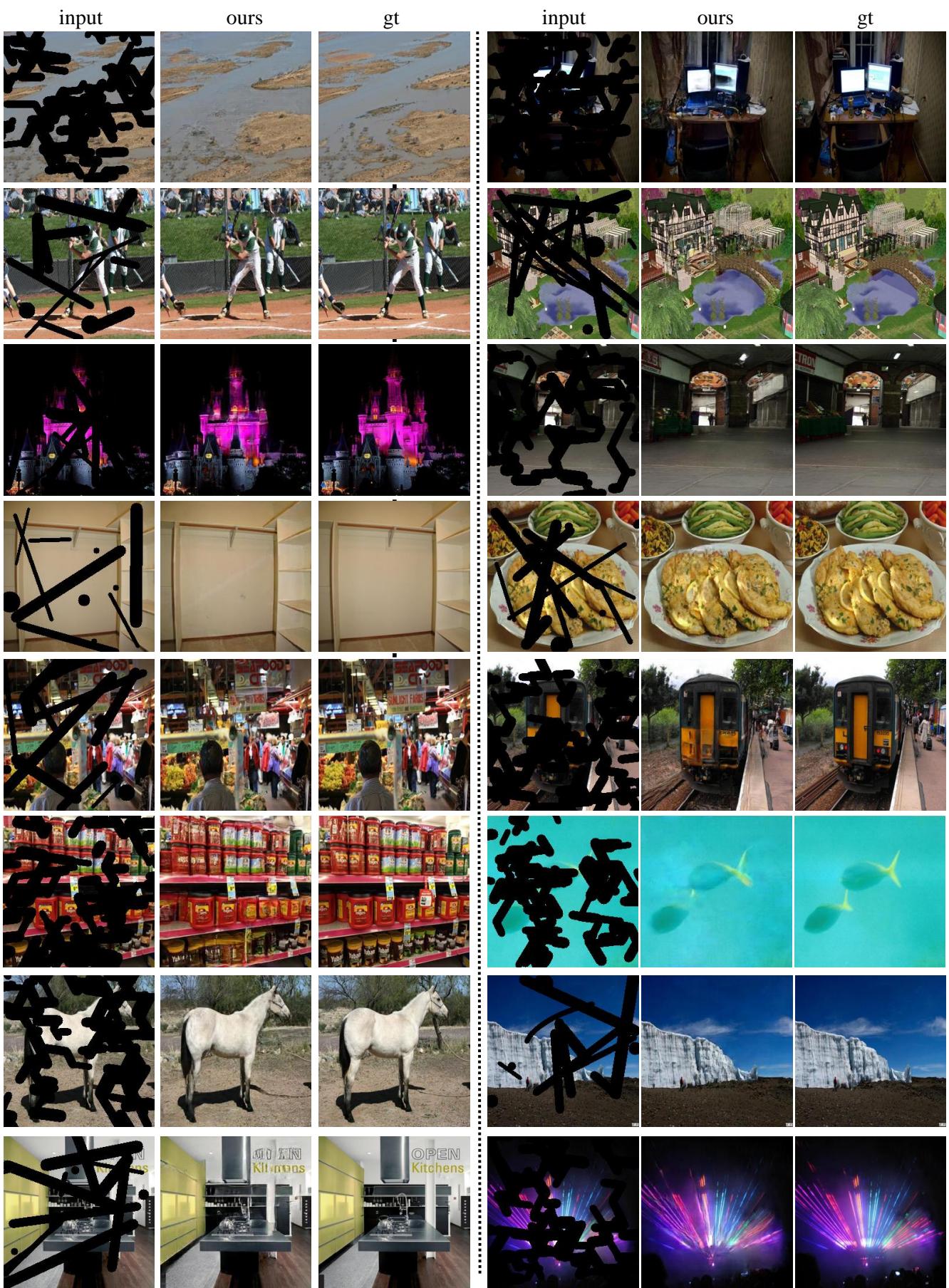


Figure 9: Qualitative results on Places2.



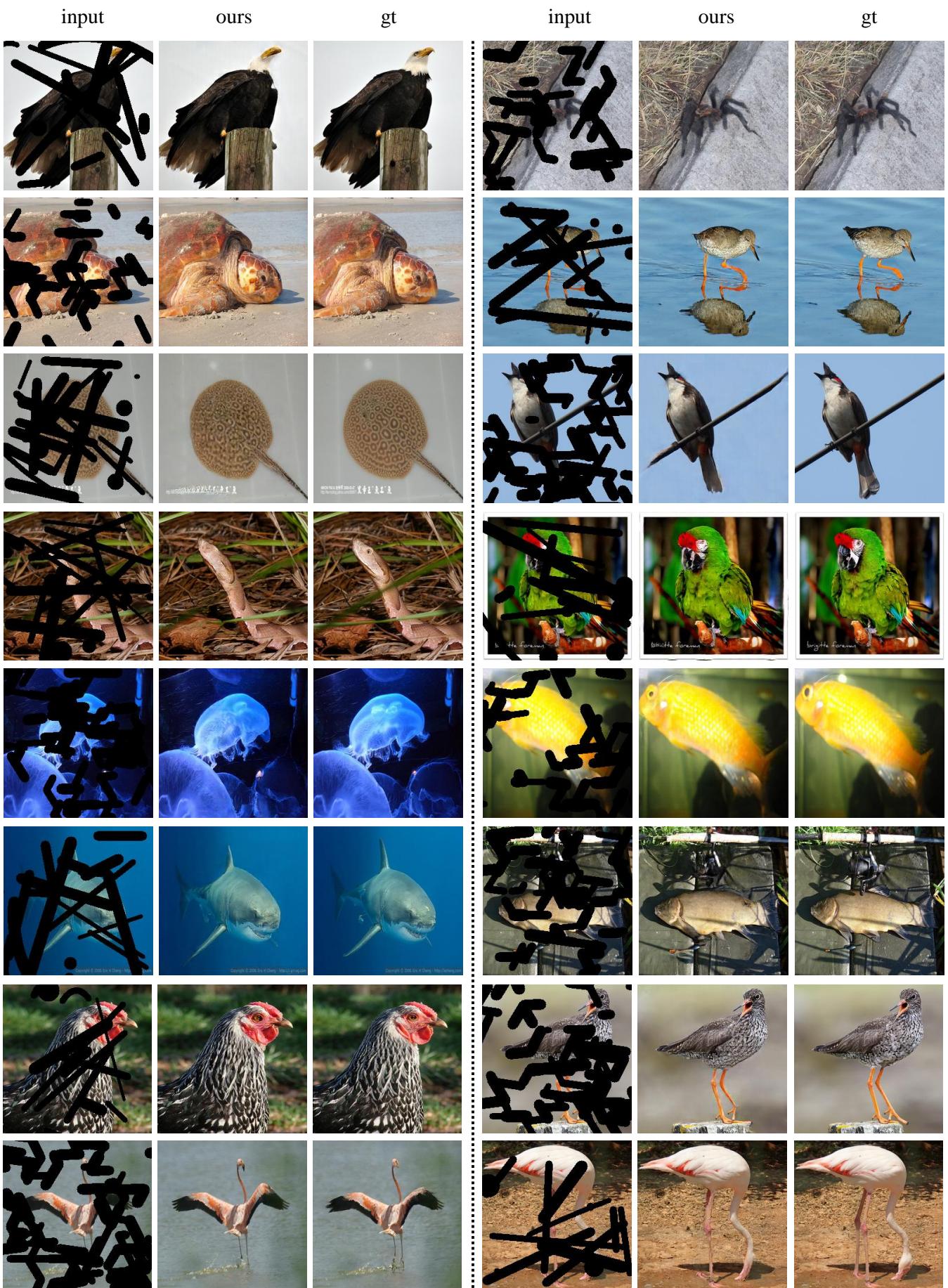


Figure 11: Qualitative results on ImageNet-100.