SCIENCE CHINA
Information Sciences

• Supplementary File •

Orpaint: a zero-shot inpainting model for oracle bone inscription rubbings with visual mamba block

Zijie Meng[1], Yuanze Zeng[2], Xiang Chang[3], Tianshuo Xu[4], Fei Chao[2,3*], Xixin Cao[1*], Changjing Shang[3] & Qiang Shen[3]

[1]*School of Software and Microelectronics, Peking University, 102206, China;*
[2]*School of Informatics, Xiamen University, 361005, China;*
[3]*Faculty of Business and Physical Sciences, Aberystwyth University, SY23 3DB, UK;*
[4]*AI Thrust, INFO Hub, Hong Kong University of Science and Technology (Guangzhou), 511400, China*

Appendix A DDPM preliminaries

Denoising Diffusion Probabilistic Model (DDPM) [4] is a class of latent variable models that can be approximately equivalent to cascaded VAEs. During the training phase, DDPM learns the image distribution of a given training set. In the inference phase, the model samples from a random noise vector x_T and gradually denoises it to generate a high-quality output image x_0.

Appendix A.1 Model architecture

DDPM primarily consists of two components: the forward diffusion process and the reverse denoising process. In the forward process, DDPM defines a diffusion process that converts an image x_0 into Gaussian noise $x_T \sim N(\mathbf{0}, \mathbf{I})$ over T time steps. Each step of the forward process is given by Equation A1:

$$q\left(x_t \mid x_{t-1}\right)=N\left(x_t ; \sqrt{1-\beta_t} x_{t-1}, \beta_t \boldsymbol{I}\right). \tag{A1}$$

The sample x_t is obtained by adding independent Gaussian noise with variance β_t at time step t and scaling the previous sample x_{t-1} by $\sqrt{1-\beta_t}$. Thus, the forward process can be written as Equation A2.

$$q\left(x_1, \ldots, x_T \mid x_0\right):=\prod_{t=1}^{T} q\left(x_t \mid x_{t-1}\right). \tag{A2}$$

The reverse process is primarily modeled by a neural network, which predicts the Gaussian distribution parameters $\mu_\theta(x_t, t)$ and $\Sigma_\theta(x_t, t)$ to recursively sample x_{t-1} from x_t until x_0 is generated. Each step of the reverse process can be modeled as shown in Equation A3:

$$p_\theta\left(x_{t-1} \mid x_t\right)=N\left(x_{t-1} ; \mu_\theta\left(x_t, t\right), \Sigma_\theta\left(x_t, t\right)\right). \tag{A3}$$

Appendix A.2 Training objective

The training objective of the diffusion model is similar to that of VAEs, as both are likelihood-based generative models aiming to maximize the likelihood function $p_\theta(x_0)$. Due to the complexity of the posterior distribution of the original data, which is difficult to solve directly, these methods often minimize the variational lower bound (ELBO) A4 to achieve the training objective [7].

$$\begin{aligned}\mathbb{E}\left[-\log p_\theta\left(\mathbf{x}_0\right)\right] & \leqslant \mathbb{E}_q\left[-\log \frac{p_\theta\left(\mathbf{x}_{0: T}\right)}{q\left(\mathbf{x}_{1: T} \mid \mathbf{x}_0\right)}\right] \\ & =\mathbb{E}_q\left[-\log p\left(\mathbf{x}_T\right)-\sum_{t \geqslant 1} \log \frac{p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)}{q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)}\right]=L.\end{aligned} \tag{A4}$$

According to the derivation in DDPM [4], this variational lower bound can be expressed as the expectation of the sum of three parts, as shown in Equation A5:

$$\begin{cases} L_{\mathrm{vlb}} := L_0+L_1+\ldots+L_{T-1}+L_T, \\ L_0 := -\log p_\theta(x_0 \mid x_1), \\ L_{t-1} := D_{KL}(q(x_{t-1} \mid x_t, x_0) \| p_\theta(x_{t-1} \mid x_t)), \\ L_T := D_{KL}(q(x_T \mid x_0) \| p(x_T)) \end{cases} \tag{A5}$$

* Corresponding author (email: fchao@xmu.edu.cn, cxx@ss.pku.edu.cn)

In this formulation, the L_T term consists of constants and does not require optimization. The L_0 term can be approximated by solving the difference of continuous function integrals. The focus is on optimizing the L_{t-1} term, i.e., parameterizing $q(x_{t-1} \mid x_t, x_0)$. Following the approach in DDPM [4], since all distributions in Equation A1 can be regarded as Gaussian distributions, Equation A3 can be solved in closed form using its mean and variance. In DDPM, the variance is a predefined constant, and the mean is given by Equation A6 [4]:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right). \tag{A6}$$

Following the derivation in DDPM [4], the model can be simplified to noise prediction, with the training objective being to optimize the MSE loss for noise prediction, defined by:

$$L_{\text{simple}} = E_{t,x_0,\epsilon} \left[\|\epsilon - \epsilon_\theta (x_t, t)\|^2 \right]. \tag{A7}$$

Based on this training objective, the model can learn the parameter set θ. During the sampling process, it progressively samples noise according to a Gaussian distribution, predicts the noise added at each time step based on the values of θ, and gradually removes it to ultimately generate the noise-free posterior distribution x_0, thereby producing the target image.

Appendix B Pseudocode

Algorithm B1 Inpainting by Orpaint

Input: $x_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
for $t = T$ **to** 1 **do**
  for $u = 1$ **to** U **do**
    if $t > 1$ **then**
      $\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
      $x_{t-1}^{\text{known}} = \sqrt{\bar{\alpha}_t} x_0 + (1 - \bar{\alpha}_t)\epsilon$
      $z \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
      $x_{t-1}^{\text{unknown}} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(x_t, t) \right) + \sigma_t z$
      $x_{t-1} = m \odot x_{t-1}^{\text{known}} + (1 - m) \odot x_{t-1}^{\text{unknown}}$
      if $u < U$ **and** $t > 1$ **then**
        $x_t \sim \mathcal{N}\left(\sqrt{1-\beta_{t-1}} x_{t-1}, \beta_{t-1} \boldsymbol{I} \right)$
      end if
    else
      $\epsilon = \mathbf{0}$
      $z = \mathbf{0}$
    end if
  end for
end for
Return: x_0

This pseudocode describes an inpainting algorithm called **"Orpaint."** The process starts with an input noise vector,

$$x_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}),$$

sampled from a standard normal distribution. The algorithm iteratively denoises x_T over a series of timesteps t, running from T down to 1. For each timestep t, an inner loop executes U iterations. During each iteration:

1. If $t > 1$, a noise vector $\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ is sampled, and the known part of x_{t-1}, denoted as x_{t-1}^{known}, is computed using the formula:

$$x_{t-1}^{\text{known}} = \sqrt{\bar{\alpha}_t} x_0 + (1 - \bar{\alpha}_t)\epsilon.$$

2. Similarly, the unknown part, x_{t-1}^{unknown}, is computed as:

$$x_{t-1}^{\text{unknown}} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(x_t, t) \right) + \sigma_t z,$$

where $z \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.

3. The final x_{t-1} is a combination of the known and unknown parts, weighted by a binary mask m, such that:

$$x_{t-1} = m \odot x_{t-1}^{\text{known}} + (1 - m) \odot x_{t-1}^{\text{unknown}}.$$

4. If $u < U$ and $t > 1$, the algorithm resamples x_t based on a normal distribution with:

$$\text{mean} = \sqrt{1-\beta_{t-1}}x_{t-1}, \quad \text{variance} = \beta_{t-1}\boldsymbol{I}.$$

5. When $t = 1$, the noise terms ϵ and z are set to zero.
Finally, the algorithm returns the denoised output x_0.

Appendix C Supplementary experimental details

Appendix C.1 Experimental setup

Given the limited scale of the Oracle bone inscription rubbing image dataset, we cannot rely solely on this dataset for model pretraining. Therefore, we combined the ImageNet dataset with the Oracle bone inscription rubbing dataset to pretrain the Denoising Diffusion Probabilistic Model (DDPM). This approach allows the pretrained model weights to encompass both fundamental image semantic features and specific distribution preferences from the ImageNet dataset. During the DDPM pretraining process, we followed the methodology of Guided Diffusion [2], randomly extracting 5% of subsets from ImageNet1K and the Oracle bone inscription image dataset to pretrain the diffusion model. To increase the proportion of Oracle bone inscription images in the training set, we applied six data augmentation methods: random angle rotation, scaling, cropping, flipping, translation, and elastic deformation [13]. All images were scaled to 256×256 for training. The training hyperparameters were set as in Repaint [10], and we used AdamW as the optimizer, adjusting its learning rate according to the inverse square root of the channel multiplier to accommodate different model sizes. Training was conducted over approximately 150,000 iterations on four RTX 3090 GPUs for seven days.

All model parameters were counted using PyTorch 2.1.2's torch.nn.Module.parameters() function, GFLOPs were calculated using the thop library of the same version, and the average time to inpaint a single image was measured using the tqdm library of the same version. For evaluating the performance of Repaint and Orpaint, we used $T = 250$ time steps and performed $r = 10$ resampling steps with a jump size of $j = 10$. For evaluating DPS and DDRM, we used the settings specified in their respective original papers.

Appendix C.2 Masking strategies and evaluation metrics

Appendix C.2.1 *Masking strategies*

By analyzing 16,000 images from the Oracle bone inscription rubbings dataset used in this study, we have roughly categorized these images into three distinct classes based on the degree of degradation: fine-grained degradation, coarse-grained degradation, and extensive degradation. For these three types of degraded images, we employed three different masking strategies during model training and evaluation. For fine-grained degraded images, we used fine masks; for coarse-grained degraded images, we used broad masks; and for large-area degraded images, we made several attempts, including box masks inspired by LaMa [15], expand64 masks, Genhalf masks, and nn2 masks inspired by SR3 [14]. The specific masking strategies are illustrated in Figure C1.

Appendix C.2.2 *Evaluation metrics*

Inpainting quality

(1) LPIPS LPIPS (Learned Perceptual Image Patch Similarity) [17] measures image similarity using features from deep learning networks, simulating human visual perception. Unlike traditional methods, it uses features from pretrained networks (e.g., AlexNet) and an optimized linear layer for human perception. In image inpainting, lower LPIPS values indicate that the inpainted image is perceptually closer to the original, reflecting better image restoration quality.

(2) FID FID (Fréchet Inception Distance) [3] measures the fidelity and diversity of generated images, often used in evaluating diffusion models. FID uses a pretrained Inception network to extract features and models these with a multivariate Gaussian distribution. The Fréchet distance between the distributions of generated and real images quantifies their similarity. Lower FID values indicate that the generated images' distribution is closer to that of real images, implying higher quality and diversity.

Computational cost In terms of computational cost, following previous research [9], we primarily used parameters, GFLOPs, and the average time to inpaint a single image to measure the complexity and resource consumption of each model. All models were evaluated based on a 256×256 image resolution.

Appendix C.3 Comparative experiments

In this section, as described in section Appendix C.2.1, we applied the three types of masks to our inpainting process. To emphasize the role of integrating the VSS block into the denoising network and to minimize the differences caused by different model principles, we selected models with similar underlying principles for comparison. Specifically, we selected the Repaint, which is the source of the Orpaint diffusion architecture, for projection-based methods; the DDRM model [19] for decomposition-based methods; and the DPS model [1] for posterior estimation-based methods. Notably, we did not choose some supervised methods for comparison, such as DSI [12] based on autoregressive models, CoModGAN [20] based

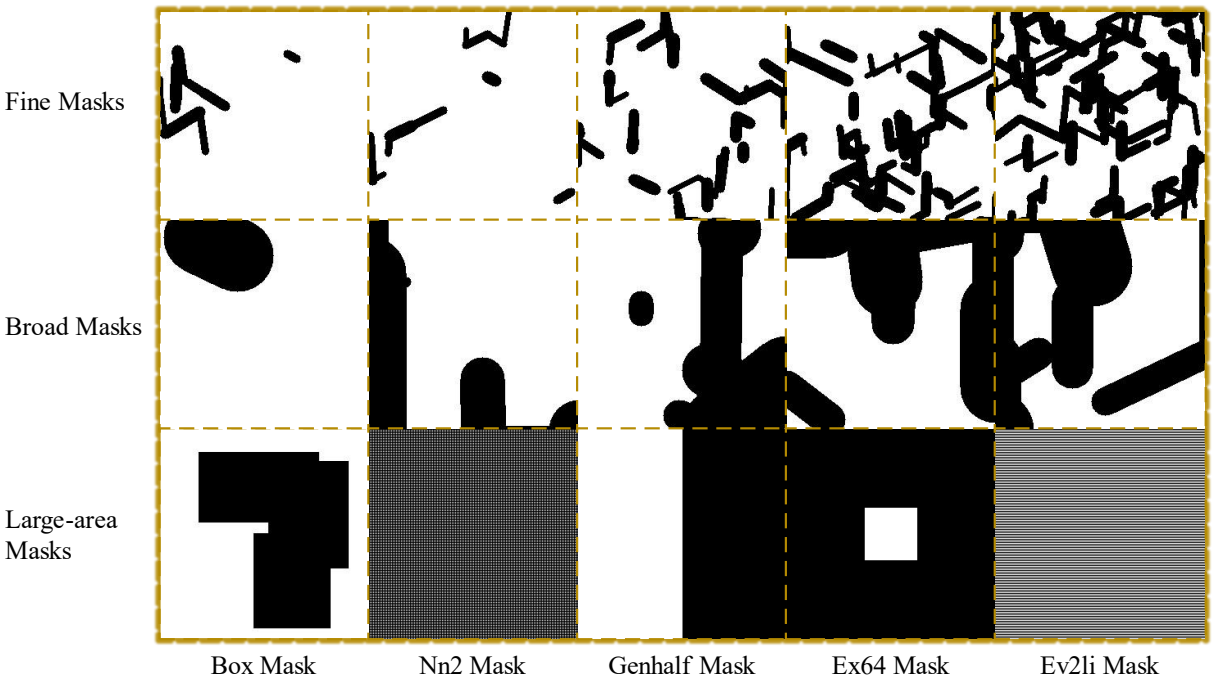

Figure C1 Three types of masks used in the experiments. The first row represents fine masks with increasing coverage from left to right, the second row represents broad masks, and the third row represents large-area masks, including box masks, nn2 masks, genhalf masks, expand64 masks, and ev2li masks from left to right.

on GANs, and LaMa [15] which performs well with large-area masks. This is mainly due to the limited number of training images in the current Oracle bone inscription dataset, totaling only 16,000 images, making it difficult to train a powerful autoregressive model or a well-converging GAN network. Based on experience, even if the models can be trained, their generalization and robustness are relatively poor, so we directly abandoned supervised learning-based image inpainting models.

To ensure fair comparisons, we used the same masks in the same tasks and pretrained the base models and configured hyperparameters as described in section Appendix C.1. We compared the inpainting effects of these zero-shot methods qualitatively and quantitatively, with the results as follows.

Appendix C.3.1 *Qualitative comparison*

Figures C2 and C3 compare the inpainting results generated by different zero-shot models. From the perspective of different masks, we can draw the following conclusions:

Fine masks The inpainting effects under fine masks are shown in Figure C2. From the perspective of masks, fine masks exhibited the best inpainting effects among all masks. The results achieved a good balance between diversity and fidelity. We believe that the variance factor randomly introduced during sampling is crucial for achieving excellent diversity. Simultaneously, the good fidelity is attributed to the resampling process and the efficient state-space equations employed by the VSS block. In model comparisons, DDRM and DPS exhibited some degree of artifacts in the inpainting results, especially near the center areas of the masks. Comparatively, Repaint and the proposed Orpaint models demonstrated more ideal inpainting effects in these areas. Qualitative analysis indicates that among the four models compared, the Orpaint model achieved the most satisfactory inpainting results under fine masks.

Broad masks The inpainting effects under broad masks are shown in Figure C3. To highlight key points, only the inpainting results of the Orpaint model are presented. We observed that under the broad masks, the fidelity of inpainting was slightly lower than that of fine masks, but the diversity was slightly higher. This phenomenon aligns with our theoretical prediction, namely that larger mask areas usually entail less prior information and greater freedom in inpainting.

Large-area masks The inpainting effects under large-area masks show significant differences among various masks, as illustrated in Figure C3. Nn2 masks and ev2li masks, due to the continuity of their masked parts, did not severely disrupt the semantic information of the image, thereby achieving high fidelity but poor diversity. In contrast, the inpainting effects of box masks were more random, highly dependent on the specific distribution of the masks. ex64 and genhalf masks covered large amounts of image information, causing the model to overly rely on the distribution characteristics of the pre-trained dataset during the generation process, thereby increasing the risk of mode collapse.

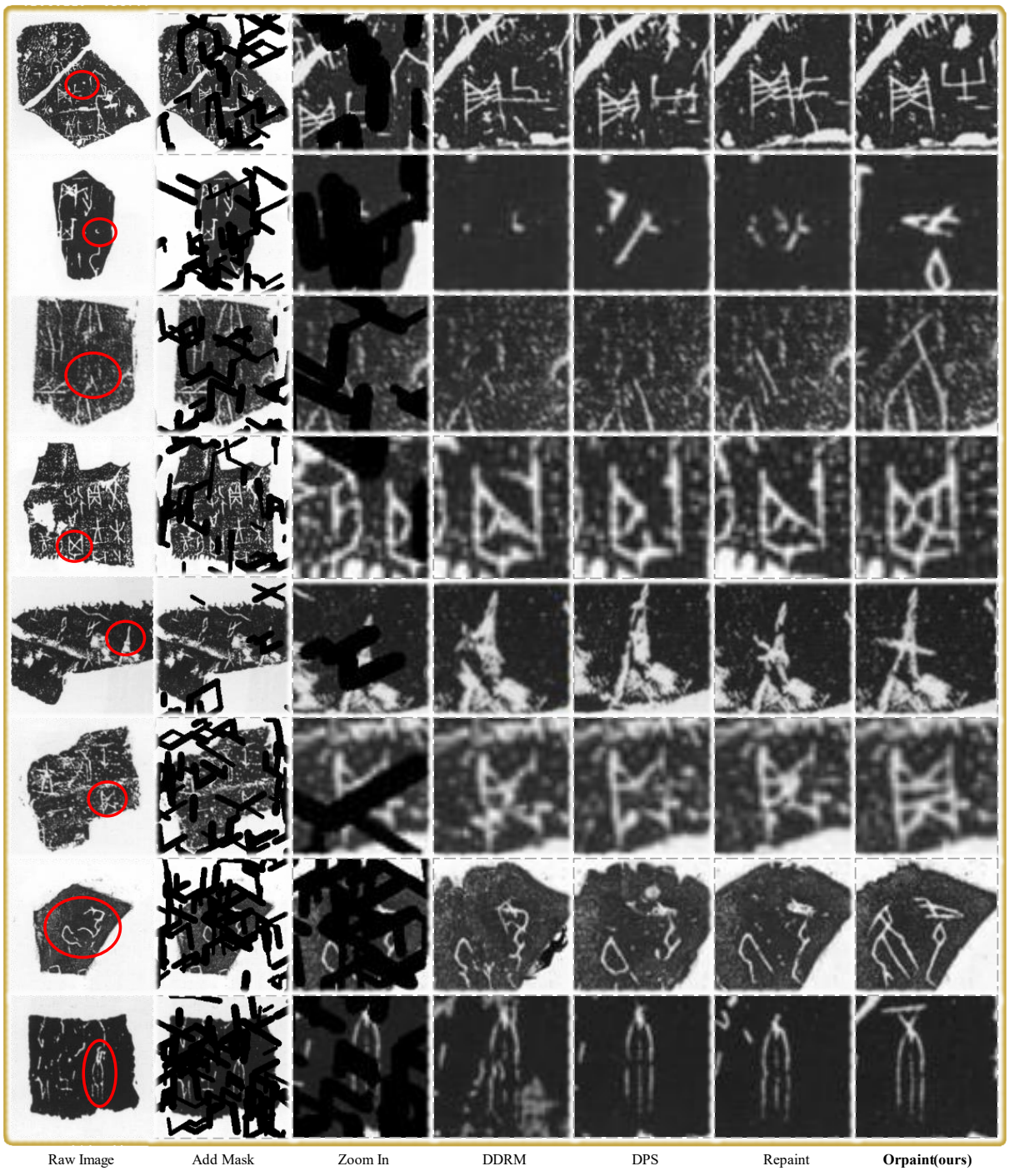

Figure C2 Qualitative inpainting effects with fine masks

Appendix C.3.2 *Quantitative comparison*

This section evaluates the Orpaint architecture in two quantitative aspects: the inpainting effect and the time and computational resources consumed for inpainting images. The metrics for each model under each mask type are derived from the average of 1,000 randomly selected inpainting results, except for large-area masks, where 200 samples were randomly selected for each mask type, and the comprehensive average was calculated. The specific results are detailed in Table C1.

Analyzing Table C1, it is evident that in terms of inpainting quality, Repaint, and Orpaint achieved the best performance under various large-area mask types. Specifically, when the mask area was small, Repaint, which incorporates multi-head self-attention blocks, exhibited more stable performance. As the mask area increased, Orpaint, which integrates the VSS block, progressively improved its performance, even surpassing Repaint, consistent with our qualitative results. In terms of computational resources and time costs, DDRM utilized matrix SVD decomposition techniques that can be accelerated

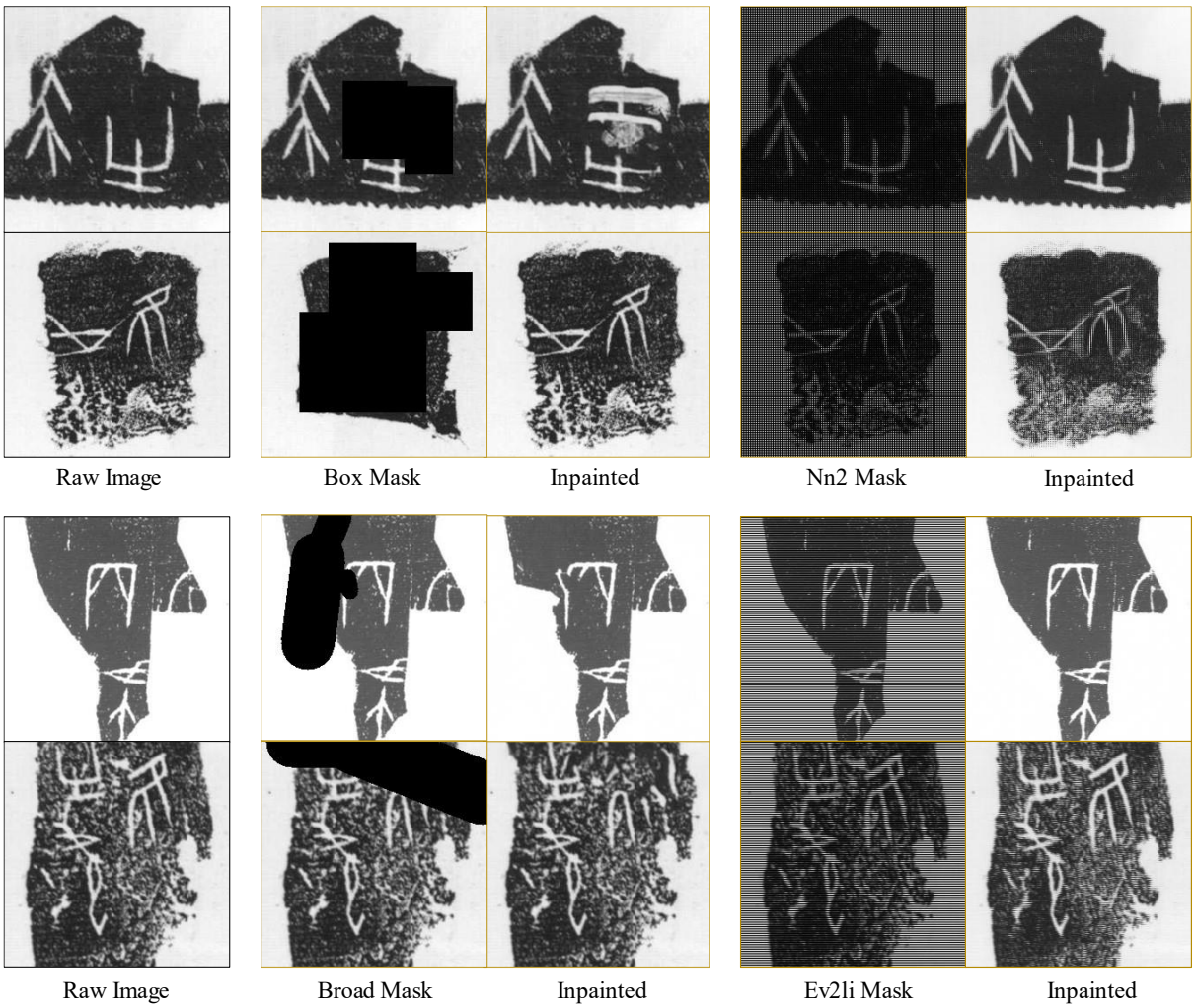

Figure C3 Qualitative inpainting effects with other masks

by parallelization, resulting in the shortest time consumption compared to other methods. Repaint, due to extensive self-attention calculations and residual connections, consumed the most time. Although Orpaint did not surpass DDRM in terms of time consumption, it outperformed Repaint in terms of model complexity, parameter count, and time consumption, demonstrating that our improved U-Net indeed accelerated the inpainting process.

Appendix C.4 Ablation studies

Appendix C.4.1 *Resolution expansion and U-Net structure ablation*

Due to the introduction of the VSS block, Orpaint achieves significant breakthroughs in long-sequence modeling. Therefore, we conducted expansion experiments on input image resolution to assess Orpaint's computational efficiency and inpainting quality as the input image size increases, verifying whether it truly inherits the advantages of VMamba. Specifically, we used unsupervised methods [19] to upscale the original input size of 256×256, obtaining images with resolutions of 512×512 and 1024×1024 to simulate higher-definition Oracle bone inscription rubbings. The experimental setup in section Appendix C.1 was used to evaluate Orpaint's inpainting quality and time consumption at different input image resolutions. This section also introduces a control group, using Repaint with a conventional U-Net structure, under the same conditions to compare the scalability of the two models.

From Figures C4, it is evident that in terms of computational resources and time costs, measured by FLOPs, Orpaint's computational complexity grows linearly, comparable to architectures based on CNN and Linear Attention. This aligns well with the theoretical conclusions of selective SSM, which we attribute mainly to the successful introduction of global 2D prior information through linear scanning by the Cross-Scan design [8]. Conversely, Repaint's complexity grows polynomially. Particularly, when the resolution reaches 1024×1024, Repaint struggles to inpaint an image within an acceptable time frame, whereas Orpaint does not. In terms of inpainting quality, both models maintain relatively stable quality without noticeable artifacts or mode repetition, indicating that both models capture the details introduced by higher image resolutions well. Although Orpaint's performance slightly lags behind Repaint as the resolution increases, the difference is minimal. Considering time consumption and computational complexity, Orpaint demonstrates better scalability. Table C2

Table C1 Inpainting effects and computational resources and time costs. Bold indicates optimal performance, while underline indicates suboptimal performance.

Mask	Model	Inpainting Effect		Comp. Resources & Time Costs		
		LPIPS↓	FID↓	Time(s/img)↓	#Para	FLOPs(G)
Fine Mask	Masked Image	0.19	78.85	-	-	-
	DDRM [19]	0.09	9.01	**14.2**	-	1220.28
	DPS [1]	0.11	16.79	<u>179.2</u>	-	1220.28
	Repaint [10]	**0.04**	**8.42**	379.5	179M	4110.41
	Orpaint (ours)	<u>0.06</u>	<u>8.81</u>	195.5	33M	1005.10
Coarse Mask	Masked Image	0.35	165.12	-	-	-
	DDRM	0.11	40.39			
	DPS	0.18	99.58			
	Repaint	**0.07**	**21.45**			
	Orpaint (ours)	<u>0.08</u>	<u>28.12</u>			
Large-area Mask	Masked Image	0.65	291.55	-	-	-
	DDRM	0.29	115.15			
	DPS	0.35	214.39			
	Repaint	<u>0.19</u>	<u>79.24</u>			
	Orpaint (ours)	**0.14**	**71.14**			

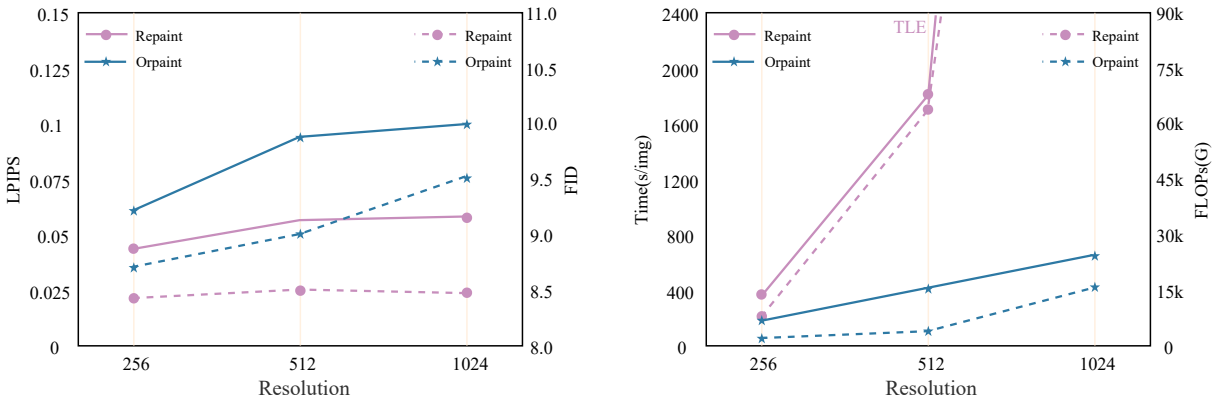

Figure C4 Results of resolution expansion and U-Net structure ablation experiment. Left side shows comparison of restoration effects; Right side displays comparison of model complexity and runtime.

Table C2 Ablation study results. Red font represents expenses that are not sustainable, blue font represents feasible expenses.

Model	Image Resolution	#Para	Time(s/img)	FLOPs(G)	LPIPS↓	FID↓
Repaint (without VSS)	256×256	179M	379.5	4,110.41	0.04	8.42
	512×512	179M	1,829.4	66,716.56	0.05	8.44
	1024×1024	179M	**8,000+**	**>1e6**	0.05	8.41
Orpaint (with VSS)	256×256	33M	195.5	1,005.10	0.06	8.81
	512×512	33M	395.2	4,012.58	0.09	9.01
	1024×1024	33M	669.1	16,448.32	0.10	9.56

provides more detailed data on the performance changes of the two models at different resolutions.

Table C3 Ablation study results for 2D scanning modes. Bold indicates optimal performance, while underline indicates suboptimal performance.

2D Scanning Mode	LPIPS↓	FID↓	Time(s/img)↓
MSA [10]	[0.047, **0.048**]	[**8.404**, **8.422**]	379.5
BiDirectional Scan [18]	[0.060, 0.067]	[8.797, 8.912]	<u>210.7</u>
Cross-Scan [8]	[0.053, 0.057]	[8.766, 8.801]	267.3
Local Scan [6]	[**0.046**, 0.055]	[<u>8.408</u>, 8.656]	391.3
Continuous Scan [16]	[0.052, 0.056]	[8.792, 8.801]	278.2
Zigzag Scan [5]	[<u>0.051</u>, <u>0.053</u>]	[8.515, <u>8.520</u>]	351.5
ES2D [11]	[0.057, 0.064]	[8.774, 8.808]	**195.5**

Appendix C.4.2 *2D scanning mode abalation*

This section presents an ablation study conducted to investigate whether the key component in reducing time overhead lies in the 2D scan technique and whether different 2D scanning methods significantly impact the time overhead and inpainting performance of the Orpaint model. The experiments compared the performance of various 2D scan methods, including BiDirectional Scan [18], Cross-Scan [8], Local Scan [6], Continuous Scan [16], Zigzag Scan [5], and ES2D [11], reconstructing the VSS block to highlight their differences and advantages over traditional models based on multi-head self-attention (MSA) [10]. The experimental setup mirrors that of Section Appendix C.1.

Given the minimal performance differences among the various 2D scanning methods, ten trials were conducted on a randomly selected sample size of 1000 from the overall dataset to ensure a clear depiction of their performance differences and the stability of the inpainting results. The performance for each experiment was averaged, and the results of the ten trials were displayed using a closed interval. Due to the extensive volume of experiments and considering time and resource consumption, the experiments were conducted only at a resolution of 256×256. To achieve optimal results, we used the fine mask that showed the best inpainting performance as discussed in Section Appendix C.3. The experimental results are presented in Table C3.

Analyzing the results, it was observed that the ES2D scan method adopted by Orpaint achieved the shortest time overhead among all the scanning methods. Regarding inpainting performance, models based on MSA (Repaint) and Local Scan (Localmamba) performed well, indicating that the feature extraction methods that integrate local and global features have become strong contenders against traditional MSA. Although Orpaint's performance was slightly lacking, the performance gap was minimal, and this minor difference is acceptable in the specific application of Oracle bone inscription image inpainting.

It is noteworthy that most Mamba-based models, despite achieving lower resource consumption in various downstream tasks, including Oracle bone inscription image inpainting, exhibited significant performance fluctuations. Among these, only the Zigzag Scan-based model could rival the models based on MSA or Transformer, possibly due to the scanning method's enhanced local feature capture and reduced wide-range attention shift. Overall, Transformer's stability in large-scale tasks remains its irreplaceable charm, whereas Mamba is more suited for designing lightweight models where high precision is not paramount.

Appendix C.5 Generalization experiments

Although Orpaint was primarily used for inpainting Oracle bone inscription rubbings in this paper, its powerful zero-shot generation capability makes it applicable to any dataset, demonstrating strong generalization performance. To prove that the mode collapse is not a flaw of the Orpaint architecture but stems from insufficient pretraining datasets, we conducted a set of generalization experiments applying Orpaint to a subset of ImageNet. Since Guided Diffusion [2] has released weights pretrained with 100% ImageNet21K for 250,000 iterations, we directly used these weights for the inpainting task on the ImageNet subset. The qualitative results are shown in Figure C5.

Analyzing Figure C5, it is evident that with sufficient pretraining, Orpaint can produce high-fidelity and diverse high-quality images under various masks, demonstrating its robust generalization capability and emphasizing the importance of pretrained models in zero-shot generation tasks.

References

1 Chung H, Kim J, Mccann M T, et al. Diffusion posterior sampling for general noisy inverse problems. arXiv preprint arXiv:2209.14687. 2022.
2 Dhariwal P, Nichol A Q. Diffusion models beat gans on image synthesis. arXiv preprint arXiv:2105.05233. 2021.
3 Heusel M, Ramsauer H, Unterthiner T, et al. Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems. 2017.30.
4 Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models. Advances in neural information processing systems. 2020.33:6840–6851.
5 Hu V T, Baumann S A, Gui M, et al. Zigma: zigzag mamba diffusion model. arXiv preprint arXiv:2403.13802. 2024.
6 Huang T, Pei X, You S, et al. Localmamba: visual state space model with windowed selective scan. arXiv preprint arXiv:2403.09338. 2024.
7 Kingma D P, Welling M. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114. 2013.
8 Liu Y, Tian Y, Zhao Y, et al. Vmamba: visual state space model. arXiv preprint arXiv:2401.10166. 2024.

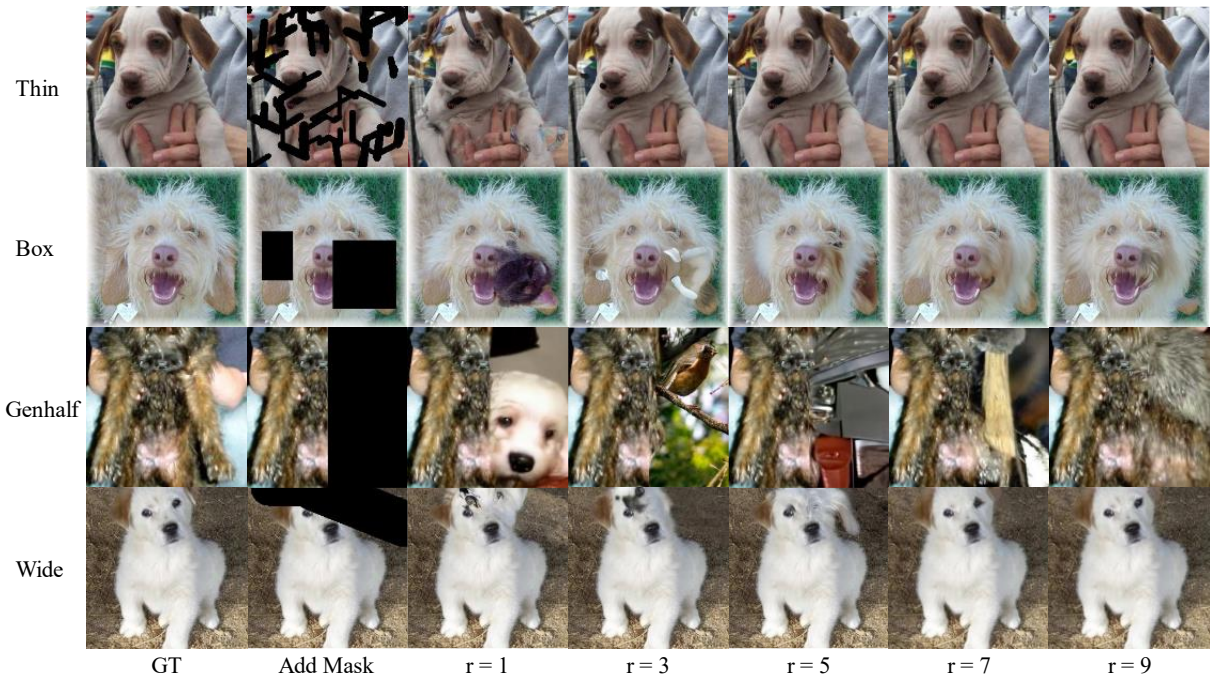

Figure C5 Generalization experiment results on ImageNet subset

9 Liu Y, Lu Y, Wei Y, et al. Research status and prospect of oracle bone inscription recognition technology. Zhishi Guanli Luntan. 2023.8(2).
10 Lugmayr A, Danelljan M, Romero A, et al. Repaint: inpainting using denoising diffusion probabilistic models. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.11461–11471.
11 Pei X, Huang T, Xu C. Efficientvmamba: atrous selective scan for light weight visual mamba. Proceedings of the AAAI Conference on Artificial Intelligence. 2025.39(6):6443–6451.
12 Peng J, Liu D, Xu S, et al. Generating diverse structure for image inpainting with hierarchical vq-vae. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.10775–10784.
13 Ronneberger O, Fischer P, Brox T. U-net: convolutional networks for biomedical image segmentation. Medical image computing and computer-assisted intervention–MICCAI 2015. 2015.18(3):234–241.
14 Saharia C, Ho J, Chan W, et al. Image super-resolution via iterative refinement. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2022.
15 Suvorov R, Logacheva E, Mashikhin A, et al. Resolution-robust large mask inpainting with fourier convolutions. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. 2022.2149–2159.
16 Yang C, Chen Z, Espinosa M, et al. Plainmamba: improving non-hierarchical mamba in visual recognition. arXiv preprint arXiv:2403.17695. 2024.
17 Zhang R, Isola P, Efros A A, et al. The unreasonable effectiveness of deep features as a perceptual metric. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.586–595.
18 Zhu L, Liao B, Zhang Q, et al. Vision mamba: efficient visual representation learning with bidirectional state space model. Forty-first International Conference on Machine Learning. 2024.
19 Kawar B, Elad M, Ermon S, et al. Denoising diffusion restoration models. Advances in Neural Information Processing Systems. 2022.35:23593–23606.
20 Zhao S, Cui J, Sheng Y, et al. Large scale image completion via co-modulated generative adversarial networks. International Conference on Learning Representations. 2021.