ZERO-SHOT BLIND INPAINTING WITH FLOW MATCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

The blind inpainting (BI) problem, a challenging variant of image inpainting, aims to reconstruct missing or corrupted regions when the corruption mask is unknown. Recent deep-learning approaches have predominantly adopted data-driven paradigms, whose generalization depends on the diversity and quality of paired training data. In this paper, we revisit zero-shot BI using pretrained generative priors. The only prior zero-shot method, RGI, explicitly optimizes a sparsity-regularized mask. We show that its solution is inherently sensitive to the unknown corruption budget and can become degenerate when the generative prior interpolates the corrupted observation. To address these limitations, we propose **ZSBI-FM**, a mask-free framework that models corruption as a structured sparse reconstruction residual. Our nested SCAD objective combines pixel-level sparsity with SLIC superpixel groups to handle both isolated and spatially coherent corruptions, while an unsupervised stopping criterion limits overfitting. Across diverse mask geometries and fillings, **ZSBI-FM** improves PSNR over the strongest competing method by 2.02–2.49 dB on FFHQ and 0.78–1.40 dB on ImageNet, and remains robust under additional measurement noise. On simulated BTAD, it also enables accurate defect localization without defective training images or pixel-level annotations.

1 INTRODUCTION

Image Inpainting is a fundamental **inverse problem (IP)** in computer vision that aims to reconstruct the missing or corrupted regions of an image. A common formulation of image inpainting assumes that the observed image is generated from an unknown clean image through a masking-based degradation operator:

$$\boldsymbol{y} = (1 - \boldsymbol{M}) \odot \boldsymbol{x} + \boldsymbol{M} \odot \boldsymbol{c} \tag{1}$$

where $\boldsymbol{y}$, $\boldsymbol{x}$, and $\boldsymbol{c}$ denote the observed corrupted image, the unknown ground-truth clean image, and the corruption signal, respectively. The term $\boldsymbol{M} \in \{0,1\}^{h \times w}$ denotes the binary mask indicating the corrupted regions. Contemporary **image inpainting** literature (Xiang et al., 2023; Quan et al., 2024) predominantly considers the conventional linear IP, in which the degradation mask $\boldsymbol{M}$ is predefined and known, which should be referred to as **Non-Blind Inpainting**. This assumption substantially simplifies the problem by fixing the forward degradation operator; However, in practical scenarios, the mask is often unknown and must be inferred jointly with the missing content. Consequently, the practically relevant task is the more challenging **Blind Inpainting (BI) problem**. Then the problem becomes a nonlinear IP where the forward function depends jointly on the unknown clean image and the unknown corruption process, the joint dependence on the unknown image and degradation mask, and multiple combinations of $(\boldsymbol{M}, \boldsymbol{x}, \boldsymbol{c})$ can lead to the same measurement $\boldsymbol{y}$.

Driven by the remarkable success of deep learning, recent studies have actively explored its adaptation to the **BI** problem. One mainstream direction is the **data-driven paradigm** (Cai et al., 2017; Liu et al., 2017; Zhao et al., 2022; Phutke et al., 2023; Wang et al., 2025; 2020; 2023b; Zhao et al., 2025). The core idea is to directly learn a mapping from contaminated images to their corresponding ground-truth counterparts, i.e., $\boldsymbol{y} \mapsto \boldsymbol{x}$, using paired training datasets $\{(\boldsymbol{y}_i, \boldsymbol{x}_i)\}_{i=1}^{N}$. While data-driven methods have demonstrated strong performance, their generalization ability depends heavily on whether the training data sufficiently cover the diversity of both image content and corruption

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

patterns encountered at test time. Constructing large-scale paired datasets with realistic corruptions and high-quality ground-truth images, however, remains challenging in many practical scenarios.

In light of these challenges, we pursue a **zero-shot approach** to solve **BI** using pretrained **deep generative priors (DGPs)** that approximate the underlying clean image manifold. Rather than relying on task-specific training data, zero-shot methods exploit the intrinsic structure captured by pretrained DGPs to regularize the solution space. As such, they provide a flexible and data-efficient alternative and are expected to serve as an important complement to data-driven approaches for practical **BI**, particularly in scenarios where relevant data are scarce, expensive, or unavailable.

The only existing zero-shot baseline, RGI (Mou et al., 2023), simultaneously optimizes the mask $\boldsymbol{M}$ with the latent seed $\boldsymbol{z}$ of the pretrained GAN to reconstruct the target image on the learned clean image manifold. Despite providing an important first step toward zero-shot **BI**, RGI inherits several limitations from both its mask-based formulation and its generative backbone. First, their **explicit mask estimation** tightly couples reconstruction quality with mask recovery, making the solution sensitive to the mask formulation and its associated hyperparameters. Moreover, **GAN models** have been widely reported to be less effective than diffusion models (DMs) and flow-matching (FM) models when solving IPs (Wang et al., 2022; Kawar et al., 2022).

In this paper, we address the aforementioned limitations of RGI by introducing a new mask-free framework, **ZSBI-FM**, for the **BI** problem. Instead of naively following the degradation process of **BI** (i.e., Eq. (1)), we treat the problem as a manifold denoising problem and model unknown corruption through structured reconstruction residuals. Here, we pair a pretrained generative prior, specifically using FMPlug (Wan et al., 2025), and introduce a group-sparsity SCAD loss objective that promotes both pixel-level sparsity and region-level sparsity across low-level vision groups. Our main contributions are:

- **Identifying and theoretically characterizing** the limitations of RGI (Mou et al., 2023) for zero-shot **BI** (Sec. 2.2)
- **Proposing ZSBI-FM**, a novel framework that builds on FMPlug (Wan et al., 2025) and combines our group-sparsity SCAD loss to solve the **BI** problem.
- **Evaluating** the framework across diverse corruption-mask patterns and additional noise conditions to assess its robustness to variations in corruption geometry and measurement noise.
- **Applying** the framework to the industrial anomaly detection problem to localize anomalous regions through reconstruction residuals.

2 Background and Related Works

2.1 Pretrained DGPs for Zero-Shot IPs

In an IP, the objective is to retrieve the unknown data $\boldsymbol{x}$ from a related measurement $\boldsymbol{y} \approx \mathcal{A}(\boldsymbol{x})$, where $\mathcal{A}(\cdot)$ is the forward function. In practice, due to insufficient measurements or an approximate forward function $\mathcal{A}$, $\boldsymbol{x}$ is usually not unique to measurement $\boldsymbol{y}$ alone. Thus, in order to obtain a unique and reliable solution, prior knowledge of $\boldsymbol{x}$ is preferred to regularize the solution. The optimization formulation for solving IPs often follows the Maximum A Posteriori (MAP) formulation (Geman & Geman, 1984; Besag, 1986):

$$\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x}} \ \ell(\boldsymbol{y}, \mathcal{A}(\boldsymbol{x})) + \Omega(\boldsymbol{x}). \tag{2}$$

where the reconstruction loss $\ell(\cdot, \cdot)$ measures the consistency of the predicted data sample with the measurement, and $\Omega(\boldsymbol{x})$ encodes prior knowledge of the underlying data.

DM/FM-Based Approaches Pretrained deep generative models on clean images provide powerful data priors for solving IPs in a zero-shot manner, without task-specific training or fine-tuning. Recent works mostly focus on pretrained DMs/FMs, which transform a simple source distribution $p_0(\boldsymbol{z})$ into a target distribution $p_1(\boldsymbol{x})$ (Lai et al., 2025; Lipman et al., 2024). Their methods can be divided into two families: the **interleaving approach** and the **plug-in approach**. Interleaving approaches (Chung et al., 2024; Mardani et al., 2023; Song et al., 2023; Pokle et al., 2023; Zhang et al., 2024; Martin et al., 2024; Kim et al., 2025) alternate between diffusion/ODE steps and iterative updates that promote measurement feasibility, i.e., $y \approx \mathcal{A}(\boldsymbol{x})$. As these two iterative processes

perturb one another, such methods are not guaranteed to produce solutions that simultaneously satisfy the generative prior (i.e., manifold feasibility) and measurement feasibility (Wang et al., 2024). Therefore, we do not further consider this family of methods in the remainder of this work.

The Plug-in Approach. This family of methods can be viewed as a continuation of the latent optimization paradigm used in GAN Inversion (Zhu et al., 2016). While GAN Inversion optimizes the latent seed of a fixed pretrained GAN generator, recent plug-in methods extend this idea to pretrained DMs/FMs by treating the entire generative process as a composite mapping $G_{\boldsymbol{\theta}}(\boldsymbol{z}) = \mathcal{R}(\mathcal{R}(\cdots \mathcal{R}(\boldsymbol{z})))$ from the source distribution to the target distribution. The unknown data $\boldsymbol{x}$ is then re-parameterized as $G_{\boldsymbol{\theta}}(\boldsymbol{z})$, and the latent seed $\boldsymbol{z}$ is optimized while the pretrained generative model remains fixed. Plugging this parameterization into Eq. (2) gives

$$\boldsymbol{z}^* \in \arg\min_{\boldsymbol{z}} \ \mathcal{L}(\boldsymbol{z}) \doteq \ell(\boldsymbol{y}, \mathcal{A} \circ G_{\boldsymbol{\theta}}(\boldsymbol{z})) + \Omega \circ G_{\boldsymbol{\theta}}(\boldsymbol{z}), \quad \text{Where} \quad x^* = G_\theta(\boldsymbol{z}^*) \tag{3}$$

since $G_{\boldsymbol{\theta}}$ is kept fixed, the recovered solution is constrained by the pretrained generative prior, while minimizing the data-fidelity term promotes consistency with the measurement (Wang et al., 2024; Zheng et al., 2025; Ben-Hamu et al., 2024; Wan et al., 2025). In the context of zero-shot **BI** solving, RGI (Mou et al., 2023) adopts GAN Inversion and therefore shares the same latent-optimization principle underlying the plug-in family. Due to this close connection, we accordingly discuss RGI together with plug-in-based methods in the remainder of this work.

2.2 What's Wrong with the Explicit Mask Formation for Solving **BI**?

To the best of our knowledge, RGI (Mou et al., 2023) is the only existing zero-shot approach for **BI**. RGI employed a pretrained GAN as a deep generative prior, leveraging the learned clean image manifold to reconstruct the image from an input corrupted at unknown locations. Following the classical **BI** approaches (Dong et al., 2012; Yan, 2013), RGI explicitly parameterizes the corrupted region as $\boldsymbol{M} \in \mathbb{R}^{h \times w}$. Here, they assume a small fraction of pixels is corrupted and impose a sparsity constraint on the mask $\boldsymbol{M}$. Resulting in the following constrained optimization problem:

$$\min_{\boldsymbol{M},\boldsymbol{z}} \quad \ell((\mathbf{1} - \boldsymbol{M}) \odot \boldsymbol{y}, (\mathbf{1} - \boldsymbol{M}) \odot G_{\boldsymbol{\theta}}(\boldsymbol{z})) \quad \text{s.t.} \quad \|\boldsymbol{M}\|_0 \leq k \tag{4}$$

where $G_{\boldsymbol{\theta}}$ denotes a pretrained GAN generator and k specifies an estimated upper bound on the number of corrupted pixels. To make the optimization more tractable, RGI further replaces the ℓ_0 constraint with an ℓ_1 penalty, yielding

$$\min_{\boldsymbol{M},\boldsymbol{z}} \quad \ell((\mathbf{1} - \boldsymbol{M}) \odot \boldsymbol{y}, (\mathbf{1} - \boldsymbol{M}) \odot G_{\boldsymbol{\theta}}(\boldsymbol{z})) + \lambda \|\boldsymbol{M}\|_1 \tag{5}$$

where the penalty parameter λ controls the trade-off between fitting the measurement and penalizing the coverage ratio of the estimated corruption mask. Despite its effectiveness, this formulation suffers from several important limitations:

Proposition 2.1 (Optimal mask for a fixed latent seed)**.** *Fix $\boldsymbol{z}$, flatten the image coordinates, and write $\boldsymbol{r} = |\boldsymbol{y} - G_{\boldsymbol{\theta}}(\boldsymbol{z})| \in \mathbb{R}^d$. For $1 \leq p < \infty$ and a binary mask $\boldsymbol{m} \in \{0,1\}^d$, define*

$$\Phi_k(\boldsymbol{z}) = \min_{\|\boldsymbol{m}\|_0 \leq k} \|(\mathbf{1} - \boldsymbol{m}) \odot \boldsymbol{R}\|_p^p. \tag{6}$$

Let $r_{(1)} \geq \cdots \geq r_{(d)}$ denote the ordered residual magnitudes. Then $\Phi_k(\boldsymbol{z}) = \sum_{j=k+1}^{d} r_{(j)}^p$. If $r_{(k)} > 0$, every minimizer uses the full budget $\|\boldsymbol{m}\|_0 = k$ and masks the k largest residual magnitudes, up to ties.

Proof. The proof is provided in Appendix A.2. □

Sensitivity of the solution to the hyperparameter k (λ). Although k is introduced as an upper bound for the mask size in Eq. (4), Proposition 2.1 shows that whenever the k-th largest residual is nonzero, the optimal mask necessarily exhausts the entire budget; i.e., exactly k pixels in the corrupted image are assigned to the estimated corruption support. This means that when k does not correspond to the actual corruption level, the mask $\boldsymbol{M}$ will either overestimate or underestimate the corrupted region. Overestimating k can admit multiple equally optimal masks, introducing ambiguity in the recovered corruption support. By contrast, an underspecified budget k leaves part of the corruption unmasked, allowing it to influence and potentially be absorbed into the reconstruction.

Consequently, for mask-based methods built on this formulation, accurately specifying the corruption budget is a necessary condition for reliable reconstruction; however, the true corruption level is generally unknown a priori in the **BI** problem.

Solving the ℓ_1-penalized form of the problem, i.e., Eq. (5), will not mitigate the issue; rather, it transfers the role of hard budget k to the penalty parameter λ. Thus, the estimated corruption support remains sensitive to a user-specified hyperparameter whose appropriate value depends on the unknown corruption level. Note that the argument above is independent of the $G_{\boldsymbol{\theta}}(\boldsymbol{z})$ term, so changing the generative prior will not eliminate this issue.

Lack of binary constraint on $\boldsymbol{M}$. Although $\boldsymbol{M}$ is intended to represent a binary corruption mask, the ℓ_1-penalized formulation does not enforce binarity for finite λ. As shown in Proposition 2.2, for any fixed latent seed $\boldsymbol{z}$, the optimal mask when using squared ℓ_2 reconstruction loss can contain entries strictly between 0 and 1. Consequently, the relaxed variable no longer directly represents corruption support but instead acts as a continuous weighting of the residuals. For this highly ill-posed inverse problem, preserving the inherent $\{0,1\}$-valued structure of the corruption support provides an additional structural constraint that rules out fractional mask solutions and more faithfully reflects the discrete nature of the underlying corruption process.

Proposition 2.2 (Non-binarity of the ℓ_1-penalized mask)**.** *Suppose that the reconstruction loss in Eq.* (5) *is the squared ℓ_2 loss. For any fixed $\boldsymbol{z}$, letting $R_{ij} = (\boldsymbol{y} - G_{\boldsymbol{\theta}}(\boldsymbol{z}))_{ij}$, the optimal mask satisfies $M^\star_{ij} = \big(1 - \lambda/\big(2R^2_{ij}\big)\big)_+$ for $R_{ij} \neq 0$, and $M^\star_{ij} = 0$ otherwise.*

Hence, $M^\star_{ij} = 0$ when $R^2_{ij} \leq \lambda/2$, whereas $0 < M^\star_{ij} < 1$ when $R^2_{ij} > \lambda/2$. Therefore, for any finite $\lambda > 0$, the ℓ_1 penalty promotes sparsity but does not enforce a binary mask; active entries approach 1 only as $\lambda \to 0$.

Proof. The proof is provided in Appendix A.3. □

These observations suggest that the central difficulty is not in selecting a better regularizer for $\boldsymbol{M}$, but rather that it is rooted in the explicit mask parameterization based on Eq. (1). Motivated by this observation, we introduce **ZSBI-FM**, a mask-free approach to robustly solve **BI**.

3 METHODS

In this section, we introduce a new framework, **ZSBI-FM**, to solve the **BI** problem in zero-shot, following the plug-in approach of fmplug (Wan et al., 2025). Instead of explicitly estimating a mask for the corrupted region, we directly focus on formulating a structural loss based on the residual between the corrupted image and the reconstructed image. In Sec. 3.1, we propose the pixel level SCAD loss that promotes pixel level sparsity. Building on this formulation, in Sec. 3.2, we further introduce our group-sparsity SCAD loss, which adapts another group-level SCAD loss that aggregates the pixel-wise SCAD responses over spatial groups, thereby promoting structured sparsity. Finally, because the expressive generative prior can gradually overfit the corrupted observation during optimization, we adopt an early-stopping strategy from prior work in Sec. 3.3 to mitigate this overfitting issue.

3.1 PIXEL-LEVEL SCAD LOSS

To address the limitation introduced by explicit mask estimation in RGI, we aim for a mask-free method formulation of the **BI** problem. Instead of following the masking-based degradation operator of Eq. (1), we reformulate the degradation as sparse additive corruption:

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{M} \odot (\boldsymbol{c} - \boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{n}, \quad \text{where} \quad \boldsymbol{x} \in \mathcal{M} \tag{7}$$

where $\boldsymbol{y}$ is the measurement, $\boldsymbol{x}$ is the clean image lying on the manifold $\mathcal{M}$, and $\boldsymbol{n} = \boldsymbol{M} \odot (\boldsymbol{c} - \boldsymbol{x})$ denotes the additive corruption. Under this formulation, corrupted locations are represented implicitly by the nonzero support of $\boldsymbol{n}$, eliminating the need to explicitly estimate $\boldsymbol{M}$. Under the sparse-corruption assumption, the **BI** problem can therefore be viewed as recovering an image on the clean-image manifold whose residual with respect to $\boldsymbol{y}$ is sparse.

Motivated by this formulation, we directly characterize the corruption in the residual domain:

$$\boldsymbol{R} = |\boldsymbol{y} - G_{\boldsymbol{\theta}}(\boldsymbol{z})| \in \mathbb{R}^{H \times W} \tag{8}$$

here, the generative prior $G_{\boldsymbol{\theta}}(\boldsymbol{z})$ constrains the reconstruction to the clean image manifold $\mathcal{M}$. When the reconstruction faithfully captures the underlying clean image, $R_{ij} \approx |n_{ij}|$ and the residual primarily reflect the sparse corruption $\boldsymbol{n}$. Accordingly, we seek a sparsity-inducing loss with ℓ_0-like behavior to promote sparsity in the residual.

Pixel-Level Scad Loss We adopt the smoothly clipped absolute deviation (SCAD) penalty (Fan & Li, 2001) and directly apply it to pixel-level residuals. The SCAD loss has the following formulation:

$$p_{\lambda,a}(x) = \begin{cases} \lambda|x|, & |x| \le \lambda, \\ (-x^2 + 2a\lambda|x| - \lambda^2)/(2(a-1)), & \lambda < |x| \le a\lambda, \\ (a+1)\lambda^2/2, & |x| > a\lambda. \end{cases} \tag{9}$$

the nonconvex, piecewise structure of SCAD assigns different levels of influence according to the residual magnitude; see Fig. 1. For small residuals, $|x| \le \lambda$, the penalty coincides with ℓ_1 and strongly encourages them toward zero, promoting consistency between the reconstruction and observations that are likely to belong to the clean image. For intermediate residuals, $\lambda < |x| \le a\lambda$, the penalty gradually reduces their influence as the residual magnitude increases. For sufficiently large residuals, $|x| > a\lambda$, the penalty saturates, and its gradient becomes zero, preventing large discrepancies from dominating the reconstruction objective. This will result in our **pixel-level Scad loss formulation**:

$$\min_{\boldsymbol{z}} \sum_{ij} p_{\lambda,a}(R_{ij}) \tag{10}$$

3.2 Group-Sparsity SCAD Loss

On more challenging **BI** datasets, the mask region can be substantial and tends to form contiguous spatial structures. In this regime, sparsity at the pixel-level becomes less informative. thus, optimizing the pixel-wise SCAD loss alone may not be sufficient. Therefore, additional structural information is needed to handle large and spatially coherent corruptions. This motivates us to move from independent pixel-wise reasoning toward a region-level representation, where neighboring pixels with similar local properties are treated jointly.

A natural idea here is to group the adjacent pixels with similar properties together. By treating each group as a structural unit, the resulting group-level sparsity prior can better capture spatially coherent corruption patterns while preserving the underlying regional structure. This raises two design questions: (1) **how should the image be partitioned into meaningful local groups**, and (2) **how should the sparsity prior be extended to group level**?

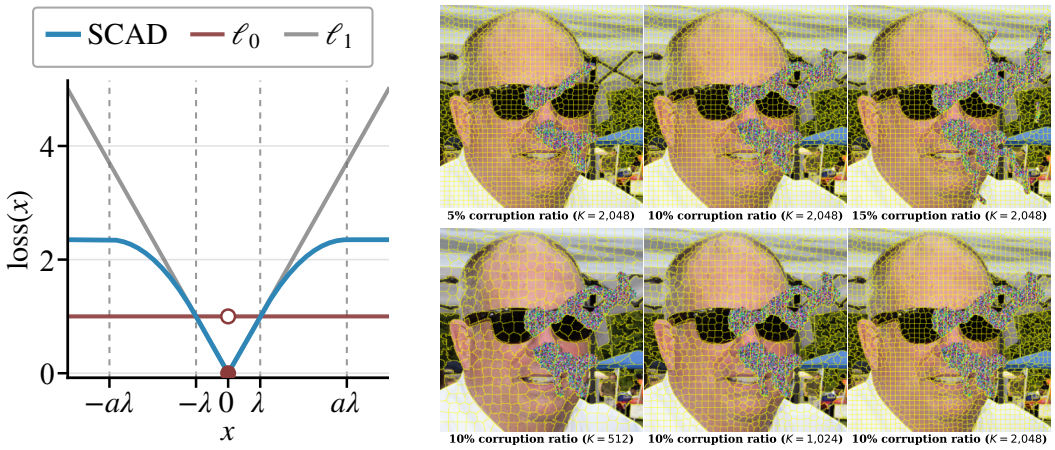


Figure 1: SCAD penalty and SLIC-based spatial grouping. Left: SCAD ($\lambda = 1, a = 3.7$) compared with ℓ_0 and ℓ_1 penalties. Right: SLIC superpixel partitions for different corruption ratios and numbers of superpixels K, used as spatial groups in the proposed group-sparsity SCAD loss.

Superpixel Segmentation To answer the first question, we revisited the Computer Vision task of Superpixel, which is a special type of image segmentation method that combines local pixels based on their low level features (Wu & Yan, 2026). This family of methods aims to over-segment an image into many homogeneous and identifiable compact partitions of similar sizes. SLIC (Achanta et al., 2012), as shown in Fig. 1, is a gold standard method for its simplicity and efficiency. It applies the k-means algorithm to the local patches and groups pixels based on their color and spatial information. We directly apply SLIC on the measurement $\boldsymbol{y}$ and produce K segments $\{\boldsymbol{S}_i\}_{i=1}^K$ as a grouping reference. Here, each $\boldsymbol{S}_i$ is a binary mask that has the same width and height as $\boldsymbol{y}$. Note that $\boldsymbol{S}_k$ is used only to provide spatial grouping for the residuals and is not intended to recover the exact corruption boundaries; therefore, moderately imperfect groupings can still provide useful local structural information.

Group-Sparsity Scad Loss Given the spatial groups $\{\boldsymbol{S}_i\}_{i=1}^K$ constructed by superpixel segmentation, our goal is to incorporate two complementary properties into the residual objective: robustness to large residuals at the pixel level and sparsity across spatial groups. Group sparsity has been extensively studied in the regression literature; in particular, group lasso (Yuan & Lin, 2006) promotes sparsity across predefined groups by penalizing group-wise aggregate energy. Beyond group level selection alone, composite penalties have also been introduced to impose penalization at both the individual and group levels (Breheny & Huang, 2009). Inspired by these ideas, we consider a nested group sparsity penalty for the residuals by integrating the aforementioned SCAD loss:

$$\min_{\boldsymbol{z}} \sum_{k=1}^{K} p_{\lambda_{group},a}(\left\|\boldsymbol{S}_k \odot p_{\lambda_{pixel},a}(\boldsymbol{R})\right\|_F/\sqrt{n_k}) \tag{11}$$

Here, we designed a nested loss where the inner and outer SCAD penalties play complementary roles. The inner penalty limits the influence of extreme residuals at the individual-pixel level while calculating the group-level energy, and the outer penalty reduces the influence of groups whose aggregate residual response is sufficiently large and promotes group-level sparsity. To balance the influence of group size, we divide the group-level energy by the square root of its corresponding group size $\sqrt{n_k}$. Solving this objective leads to our **ZSBI-FM** method.

3.3 Preventing Corruption Fitting via Early Stopping

Despite their different formulations, both RGI (as shown in Sec. A.5) and our method remain susceptible to **corruption fitting** due to the **high expressive capacity** of the DGP. In particular, when $G_{\boldsymbol{\theta}}(\boldsymbol{z})$ is sufficiently expressive, prolonged optimization can lead to overfitting, i.e. $G_{\boldsymbol{\theta}}(\boldsymbol{z}) \to \boldsymbol{y}$. In our residual formulation, this causes $\boldsymbol{R} = |\boldsymbol{y} - G_{\boldsymbol{\theta}}(\boldsymbol{z})| \to \boldsymbol{0}$, such that the sparse corruption is gradually absorbed into the reconstruction rather than remaining identifiable in the residual domain. Therefore, the generative prior alone is insufficient to prevent corruption fitting during prolonged optimization. To prevent optimization from entering this corruption-fitting regime, we adopt the early stopping strategy ES-WMV (Wang et al., 2023a; 2024) to avoid overfitting. The reconstruction is therefore taken from an intermediate stage of optimization, before the generator substantially absorbs the corrupted observations into its reconstruction. As shown in Sec. 4.2, this early stopping strategy also improves robustness to measurement noise.

4 Experiment

In Sec. 4.1, we benchmark the performance of our **ZSBI-FM** on the FFHQ (Karras et al., 2019) and Imagenet (Deng et al., 2009) datasets across multiple mask patterns. In Sec. 4.2, we further evaluate the robustness of different methods under varying noise conditions. Subsequently, in Sec. 4.3, we use anomaly detection as a downstream application to demonstrate the quality of our reconstructions. Specifically, our method effectively reconstructs the masked normal regions while preserving large residuals in anomalous regions, thereby enabling clear identification of anomalies.

4.1 Blind Inpainting with Different Corruption-Mask Patterns

Models We compare **ZSBI-FM** against several strong baselines spanning three categories. (i) **Zero-shot method:** RGI with a diffusion-model backbone (RGI-DM). For a fair comparison, both

Table 1: **Blind inpainting results on FFHQ**. Brush, Box, and Patches average equally over noise, solid-color, and DTD fillings (100 images per filling); Text uses 100 images. (**Bold**: best, under: second best, green: performance increase, red: performance decrease)

	Brush			Box			Patches			Text		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
VCNet	17.97	0.702	0.309	18.52	0.764	0.239	19.26	0.712	0.279	13.07	0.406	0.481
OmniNet	19.90	0.874	0.202	20.57	0.915	0.120	20.57	0.900	0.159	22.24	0.920	0.123
CAML	23.93	0.832	0.160	19.93	0.764	0.249	23.88	0.830	0.161	23.06	0.802	0.192
FLUX.2-klein	25.19	0.875	0.098	25.23	0.879	0.086	25.72	0.887	0.086	25.05	0.882	0.094
FLUX.2-dev	29.30	0.916	**0.058**	28.39	0.918	**0.053**	29.09	0.921	**0.053**	29.10	0.924	**0.055**
RGI-DM	29.88	0.917	0.077	28.52	0.917	0.062	29.85	0.923	0.066	31.28	0.933	0.065
ZSBI-FM (Ours)	**32.17**	**0.936**	0.066	**31.01**	**0.936**	0.054	**32.12**	**0.940**	0.058	**33.30**	**0.947**	0.056
Ours vs. Best compe.	+2.29	+0.019	+0.008	+2.49	+0.018	+0.002	+2.27	+0.017	+0.005	+2.02	+0.014	+0.001

Table 2: **Blind inpainting results on ImageNet**. Brush, Box, and Patches average equally over noise, solid-color, and DTD fillings (100 images per filling); Text uses 100 images. (**Bold**: best, under: second best, green: performance increase, red: performance decrease)

	Brush			Box			Patches			Text		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
CAML	18.94	0.669	0.347	17.46	0.644	0.377	18.61	0.663	0.352	18.95	0.659	0.359
FLUX.2-klein	22.02	0.795	0.158	22.15	0.818	0.134	22.16	0.810	0.143	22.44	0.816	0.145
FLUX.2-dev	25.32	0.857	**0.096**	24.19	0.858	**0.096**	25.04	0.856	**0.095**	25.94	0.873	**0.086**
RGI-DM	25.38	0.852	0.125	24.41	0.858	0.108	24.94	0.857	0.114	26.87	0.880	0.107
ZSBI-FM (Ours)	**26.33**	**0.873**	0.111	**25.81**	**0.878**	0.097	**26.03**	**0.877**	0.105	**27.65**	**0.893**	0.095
Ours vs. Best compe.	+0.95	+0.016	+0.015	+1.40	+0.019	+0.002	+0.98	+0.020	+0.009	+0.78	+0.014	+0.009

RGI-DM and **ZSBI-FM** employed Flux.2-klein, a lightweight conditional image-to-image generative model, as the DGP. Here, Flux.2-klein is initialized with its publicly released pretrained weights and kept frozen throughout our experiments. (ii) **Supervised learning method:** VCNet (Wang et al., 2020), OmniNet (Phutke et al., 2023), and CAML (Zhao et al., 2025); we use the corresponding pretrained models provided by the original papers. As VCNet and OmniNet don't have pretrained model on ImageNet, here we will only report their results on FFHQ dataset. (ii) **Foundation image-to-image generative models:** the lightweight Flux.2-klein and the more capable Flux.2-dev (Labs, 2025); allowing us to examine the effect of backbone capacity on direct **BI** performance.

Dataset, tasks and measurment For both the FFHQ dataset (Karras et al., 2019) and the ImageNet dataset (Deng et al., 2009), we construct test sets by randomly sampling 100 images from each dataset. To systematically investigate how the geometry and visual appearance of corrupted regions affect the performance of different methods, we evaluate each test image under a diverse set of mask shapes and filling patterns. Specifically, we consider three **mask shapes**: (i) line strokes, (ii) convex patches, and (iii) boxes. For each mask shape, we further consider three types of **fillings**: (i) Gaussian noise, (ii) constant colors, and (iii) natural textures sampled from the Describable Textures Dataset (DTD) (Cimpoi et al., 2014). In addition, we consider **text corruption** as a separate case. This controlled set of corruptions enables us to disentangle their respective effects on method performance and assess robustness across diverse corruption characteristics. For **evaluation metrics**, we use PSNR for pixel-level difference, SSIM and DISTS for structural and texture similarity, LPIPS for perceptual difference, and CLIPIQA and MUSIQ for no-reference quality metrics.

Results on Clean Data Tabs. 1 and 2 shows that **ZSBI-FM** achieves the best PSNR and SSIM in every corruption category on both FFHQ and ImageNet datasets. Supervised methods trail by 7.4–10.4 dB and are brittle to the corruption type, with CAML losing 4.0 dB from Brush to Box and VCNet dropping to 13.07 dB on Text; in contrast, **ZSBI-FM** uses no paired data and varies by at most 2.3 dB across categories. Foundation models do not close this gap either: applied directly, FLUX.2-klein reaches only 25.1–25.7 dB on FFHQ, whereas the same frozen model reaches 31.0–33.3 dB under **ZSBI-FM**, surpassing even the much larger FLUX.2-dev by up to 4.2 dB. FLUX.2-

Table 3: **Robustness and Early Stopping.** Blind inpainting on subset of our FFHQ test set with different levels of noise. We only show the **PSNR**↑ and **PSNR GAP**↓ to save space. (**Bold**: best, under: second best, green: performance increase, red: performance decrease)

	Gaussian		Impulse		Shot		Speckle	
	Low	High	Low	High	Low	High	Low	High
VCNet	7.71	6.41	14.04	10.31	7.05	6.44	8.54	7.75
OmniNet	18.30	12.81	17.66	14.62	16.30	12.44	17.60	14.15
CAML	23.32	23.46	22.82	22.96	23.39	23.51	23.22	23.23
FLUX.2-dev	23.21	21.98	23.72	22.65	22.77	21.76	23.04	22.21
RGI-DM	22.77	21.19	24.36	24.43	22.64	20.61	22.98	22.17
ZSBI-FM (Ours)	**27.13**	**25.13**	**28.18**	**27.62**	**27.12**	**24.62**	**27.48**	**26.19**
Ours vs. Best compe.	+3.82	+1.66	+3.81	+3.19	+3.73	+1.12	+4.26	+2.96
PSNR GAP↓	0.31	0.76	0.16	0.08	0.40	0.75	0.28	0.36

dev attains slightly lower LPIPS, but its lower fidelity shows that it generates plausible rather than faithful content. The comparison with RGI-DM is the most controlled one, since both methods share an identical prior and differ only in formulation. Here **ZSBI-FM** gains 0.8–2.5 dB, most on Box and least on Text; that is, the gain is largest where contiguous corruption breaks an explicit mask budget and smallest where pixel-level sparsity already suffices. The filling type shows a similar pattern: gains are largest for solid colors and smallest for natural DTD textures, where **ZSBI-FM** trails FLUX.2-dev by at most 0.36 dB in two of three settings, as such textures lie close to the natural-image manifold (**??**). Finally, although ImageNet is harder for all methods, **ZSBI-FM** retains its lead in every category without any domain-specific training.

4.2 Robustness to Unknown Measurement Noise

For this group of experiments, we use a subset of the FFHQ dataset considered in Sec. 4.1, consisting of 100 images covering all 10 mask patterns. The measurements were corrupted with four types of noise—Gaussian, impulse, shot, and speckle—at two severity levels: low (level-1) and high (level-2), following (Hendrycks & Dietterich, 2019). As shown in Tab. 3, **ZSBI-FM** achieves the highest PSNR in all eight noise settings, outperforming the best competing method by 1.12–4.26 dB. Supervised methods trained on clean corruptions degrade sharply (e.g., VCNet drops below 15 dB), while RGI-DM and FLUX.2-dev lose up to 2 dB from low to high noise under Gaussian and shot noise. Moreover, the **PSNR GAP** between the peak of the optimization trajectory and our early-stopping point is at most 0.76 dB, showing that **the proposed stopping criterion reliably avoids overfitting** to measurement noise without access to ground truth.

4.3 Industrial Application – Anomaly Detection

One may naturally ask how a zero-shot **BI** method can be applied to pixel-wise anomaly detection. Under the formulation in Eq. (1), anomalous regions can be interpreted as the unknown corruption mask $\boldsymbol{M}$ since they deviate from the distribution of normal images and can therefore be viewed as corruptions of the underlying clean image. The corresponding normal appearance is represented by the unknown clean image $\boldsymbol{x}$. Thus, pixel-wise anomaly detection can be cast as a blind inpainting problem that jointly localizes anomalous regions and recovers their underlying normal content. This perspective is particularly suitable for industrial inspection, where defect-free images are often abundant, whereas defective samples and pixel-level annotations are scarce and costly to acquire. Such data availability enables learning a deep generative prior over the normal-image domain, which can then be leveraged for zero-shot **BI** at test time.

We evaluate **ZSBI-FM** on the simulated BTAD benchmark introduced in RGI and compare it against AnoGAN (Schlegl et al., 2017) and RGI (Mou et al., 2023) as zero-shot baselines, DRAEM (Zavrtanik et al., 2021) as a supervised anomaly detection method, and FLUX.2-dev (Labs, 2025) as a large-scale deep generative model. For the zero-shot methods, we adapt a pretrained flow-matching backbone (Lipman et al., 2024) using only defect-free training images and keep the model weights fixed during evaluation. Further implementation and evaluation details are provided in Sec. A.6.

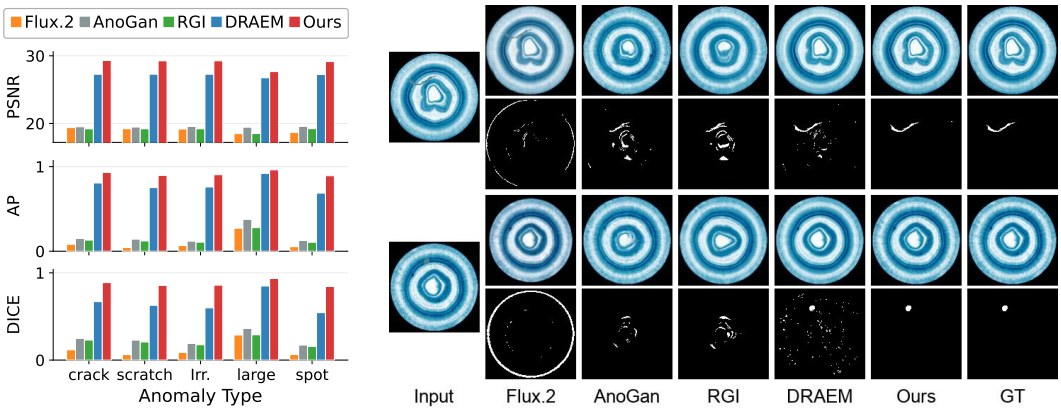


Figure 2: Quantitative (left) and qualitative (right) results for anomaly detection on simulated BTAD3 dataset. The left panel reports reconstruction quality using PSNR and anomaly-mask quality using Average Precision (AP) and Dice score. In the right panel, the first and third rows show reconstructed images, while the second and fourth rows show binary anomaly masks derived from pixel-wise ℓ_1 residuals.

Results Figure Fig. 2 shows that **ZSBI-FM** achieves the **highest reconstruction PSNR** and the best **anomaly localization performance** in terms of AP and Dice across all five defect types. Zero-shot baselines using the same flow-matching backbone, including AnoGAN and RGI, as well as FLUX.2, fail to faithfully recover the normal appearance (PSNR $\approx$ 19 dB), leading to residual maps dominated by reconstruction artifacts rather than true defects. For example, FLUX.2 alters the global texture and outer rim of the object. DRAEM, despite being trained with synthetic anomalies, produces noisy and fragmented masks, as illustrated by the spot example. In contrast, **ZSBI-FM** accurately reconstructs normal regions while preserving large residuals at defective locations, yielding anomaly masks that closely align with the ground truth. Notably, **ZSBI-FM** requires neither defective training samples nor pixel-level annotations. See Sec. A.9 for detailed results.

5 Conclusion

We presented **ZSBI-FM**, a zero-shot, mask-free framework for **BI** with pretrained generative priors. By representing unknown corruption as a structured reconstruction residual, our method avoids explicit mask estimation and combines pixel- and superpixel-level SCAD penalties with early stopping. Experiments on FFHQ and ImageNet demonstrate consistent improvements across diverse corruption patterns and measurement noise, while results on simulated BTAD show that the same formulation supports accurate industrial anomaly localization without defect annotations. These findings establish structured residual optimization as a robust alternative to explicit mask modeling for blind image restoration.

AI USE STATEMENT

In this work, we used generative AI tools to assist with formulating mathematical claims, providing critical ingredients for proving those claims, drafting and refining proofs, offering feedback on research methodology or experiments, supporting qualitative and thematic data analysis, and assisting with the interpretation of results. We have not used generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, implement the methods, assist with performing translation, clean or reformat datasets; and propose or refine research hypotheses, as this is not applicable to this work.

All AI-assisted content was subject to independent human review and verification. In particular, we carefully checked all mathematical propositions formulated with AI assistance and independently verified their proofs. We critically evaluated AI-generated feedback on experimental design and methodology and independently checked the resulting analyzes and conclusions. Where AI tools provided coding-related suggestions or feedback, we reviewed and verified the relevant code for correctness.

We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012. doi: 10.1109/TPAMI.2012.120.

Heli Ben-Hamu, Omri Puny, Itai Gat, Brian Karrer, Uriel Singer, and Yaron Lipman. D-flow: Differentiating through flows for controlled generation, 2024. URL https://arxiv.org/abs/2402.14017.

Julian Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, 48(3):259–302, 1986. ISSN 0035-9246. URL https://www.jstor.org/stable/2345426.

Patrick Breheny and Jian Huang. Penalized methods for bi-level variable selection. *Statistics and its interface*, 2(3):369–380, July 2009. ISSN 1938-7989 1938-7997. doi: 10.4310/sii.2009.v2.n3.a10.

Nian Cai, Zhenghang Su, Zhineng Lin, Han Wang, Zhijing Yang, and Bingo Wing-Kuen Ling. Blind inpainting using the fully convolutional neural network. *Vis. Comput.*, 33(2):249–261, February 2017. ISSN 0178-2789. doi: 10.1007/s00371-015-1190-z. URL https://doi.org/10.1007/s00371-015-1190-z.

Hyungjin Chung, Jeongsol Kim, Michael T. Mccann, Marc L. Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems, 2024. URL https://arxiv.org/abs/2209.14687.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Bin Dong, Hui Ji, Jia Li, Zuowei Shen, and Yuhong Xu. Wavelet frame based blind image inpainting. *Applied and Computational Harmonic Analysis*, 32(2):268–279, 2012. ISSN 1063-5203. doi: https://doi.org/10.1016/j.acha.2011.06.001. URL https://www.sciencedirect.com/science/article/pii/S1063520311000765.

Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001. doi: 10.1198/016214501753382273. URL https://doi.org/10.1198/016214501753382273.

S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE transactions on pattern analysis and machine intelligence*, 6(6):721–741, June 1984. ISSN 0162-8828. doi: 10.1109/tpami.1984.4767596.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2019. URL https://arxiv.org/abs/1812.04948.

Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models, 2022. URL https://arxiv.org/abs/2201.11793.

Jeongsol Kim, Bryan Sangwoo Kim, and Jong Chul Ye. Flowdps: Flow-driven posterior sampling for inverse problems, 2025. URL https://arxiv.org/abs/2503.08136.

Black Forest Labs. FLUX.2: Frontier Visual Intelligence. https://bfl.ai/blog/flux-2, 2025.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. URL https://arxiv.org/abs/2506.15742.

Chieh-Hsin Lai, Yang Song, Dongjun Kim, Yuki Mitsufuji, and Stefano Ermon. The principles of diffusion models. *arXiv preprint arXiv:2510.21890*, 2025.

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024.

Yang Liu, Jinshan Pan, and Zhixun Su. Deep blind image inpainting, 2017. URL https://arxiv.org/abs/1712.09078.

Morteza Mardani, Jiaming Song, Jan Kautz, and Arash Vahdat. A variational perspective on solving inverse problems with diffusion models. *arXiv preprint arXiv:2305.04391*, 2023.

S'egolene Martin, Anne Gagneux, Paul Hagemann, and Gabriele Steidl. Pnp-flow: Plug-and-play image restoration with flow matching. *ArXiv*, abs/2410.02423, 2024. URL https://api.semanticscholar.org/CorpusID:273098316.

Pankaj Mishra, Riccardo Verk, Daniele Fornasier, Claudio Piciarelli, and Gian Luca Foresti. Vt-adl: A vision transformer network for image anomaly detection and localization. In *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*, pp. 01–06, 2021. doi: 10.1109/ISIE45552.2021.9576231.

Shancong Mou, Xiaoyi Gu, Meng Cao, Haoping Bai, Ping Huang, Jiulong Shan, and Jianjun Shi. Rgi: robust gan-inversion for mask-free image inpainting and unsupervised pixel-wise anomaly detection. In *The Eleventh International Conference on Learning Representations*, 2023.

Shruti S. Phutke, Ashutosh Kulkarni, Santosh Kumar Vipparthi, and Subrahmanyam Murala. Blind image inpainting via omni-dimensional gated attention and wavelet queries. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1251–1260, 2023. doi: 10.1109/CVPRW59228.2023.00132.

Ashwini Pokle, Matthew Muckley, Ricky T. Q. Chen, and Brian Karrer. Training-free linear image inverses via flows. *Trans. Mach. Learn. Res.*, 2024, 2023. URL https://api.semanticscholar.org/CorpusID:263829001.

Weize Quan, Jiaxi Chen, Yanli Liu, Dong-Ming Yan, and Peter Wonka. Deep learning-based image and video inpainting: A survey, 2024. URL https://arxiv.org/abs/2401.03395.

Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery, 2017. URL https://arxiv.org/abs/1703.05921.

Bowen Song, Soo Min Kwon, Zecheng Zhang, Xinyu Hu, Qing Qu, and Liyue Shen. Solving inverse problems with latent diffusion models via hard data consistency. *ArXiv*, abs/2307.08123, 2023. URL https://api.semanticscholar.org/CorpusID:259937490.

Yuxiang Wan, Ryan Devera, Wenjie Zhang, and Ju Sun. Fmplug: Plug-in foundation flow-matching priors for inverse problems. *arXiv preprint arXiv:2508.00721*, 2025.

Hengkang Wang, Taihui Li, Zhong Zhuang, Tiancong Chen, Hengyue Liang, and Ju Sun. Early stopping for deep image prior. *Transactions on Machine Learning Research*, 2023a.

Hengkang Wang, Xu Zhang, Taihui Li, Yuxiang Wan, Tiancong Chen, and Ju Sun. Dmplug: A plug-in method for solving inverse problems with diffusion models. *Advances in Neural Information Processing Systems*, 37:117881–117916, 2024.

Jiahao Wang, Gang Pan, Di Sun, Jinyuan Li, and Jiawan Zhang. Afan: An attention-driven forgery adversarial network for blind image inpainting. *IEEE Transactions on Multimedia*, 27:6845–6856, 2025. doi: 10.1109/TMM.2025.3590914.

Junke Wang, Shaoxiang Chen, Zuxuan Wu, and Yu-Gang Jiang. Ft-tdr: Frequency-guided transformer and top-down refinement network for blind face inpainting. *Trans. Multi.*, 25:2382–2392, January 2023b. ISSN 1520-9210. doi: 10.1109/TMM.2022.3146774. URL https://doi.org/10.1109/TMM.2022.3146774.

Yi Wang, Ying-Cong Chen, Xin Tao, and Jiaya Jia. Vcnet: A robust approach to blind image inpainting. *arXiv preprint arXiv:2003.06816*, 2020.

Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model, 2022. URL https://arxiv.org/abs/2212.00490.

Chong Wu and Hong Yan. A survey of superpixel methods and their applications. *Signal Processing: Image Communication*, 145:117563, 2026. ISSN 0923-5965. doi: https://doi.org/10.1016/j.image.2026.117563. URL https://www.sciencedirect.com/science/article/pii/S092359652600086X.

Hanyu Xiang, Qin Zou, Muhammad Ali Nawaz, Xianfeng Huang, Fan Zhang, and Hongkai Yu. Deep learning for image inpainting: A survey. *Pattern Recognition*, 134:109046, 2023. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2022.109046. URL https://www.sciencedirect.com/science/article/pii/S003132032200526X.

Ming Yan. Restoration of images corrupted by impulse noise and mixed gaussian impulse noise using blind inpainting. *SIAM Journal on Imaging Sciences*, 6(3):1227–1245, 2013. doi: 10.1137/12087178X. URL https://doi.org/10.1137/12087178X.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(1):49–67, 02 2006. ISSN 1369-7412. doi: 10.1111/j.1467-9868.2005.00532.x. URL https://doi.org/10.1111/j.1467-9868.2005.00532.x.

Vitjan Zavrtanik, Matej Kristan, and Danijel Skocaj. Draem - a discriminatively trained reconstruction embedding for surface anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8330–8339, October 2021.

Yasi Zhang, Peiyu Yu, Yaxuan Zhu, Yingshan Chang, Feng Gao, Yingnian Wu, and Oscar Leong. Flow priors for linear inverse problems via iterative corrupted trajectory matching. *ArXiv*, abs/2405.18816, 2024. URL https://api.semanticscholar.org/CorpusID:270095081.

Haoru Zhao, Zhaorui Gu, Bing Zheng, and Haiyong Zheng. Transcnn-hae: Transformer-cnn hybrid autoencoder for blind image inpainting. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, pp. 6813–6821, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392037. doi: 10.1145/3503161.3547848. URL https://doi.org/10.1145/3503161.3547848.

Haoru Zhao, Yufeng Wang, Zhaorui Gu, Bing Zheng, and Haiyong Zheng. Context-aware mutual learning for blind image inpainting and beyond. *Expert Syst. Appl.*, 268(C), April 2025. ISSN 0957-4174. doi: 10.1016/j.eswa.2024.126224. URL https://doi.org/10.1016/j.eswa.2024.126224.

Yang Zheng, Wen Li, and Zhaoqiang Liu. Integrating intermediate layer optimization and projected gradient descent for solving inverse problems with diffusion models. *ArXiv*, abs/2505.20789, 2025. URL https://api.semanticscholar.org/CorpusID:278911348.

Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A. Efros. Generative visual manipulation on the natural image manifold. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (eds.), *Computer Vision – ECCV 2016*, pp. 597–613, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46454-1.

A Appendix

A.1 Related Works: Data-Driven **BI** Methods

Data Driven methods for solving the **BI** problem can be broadly categorized into two families: **end-to-end** and **two-stage** approaches. End-to-end methods directly learn a mapping from the contaminated image to the clean image manifold. Cai et al. (Cai et al., 2017) first introduced an End to End supervised CNN that directly maps degraded images to their clean counterparts. Liu et al. (Liu et al., 2017) trained a CNN to learn the residuals within the masked regions for restoration. Zhao et al. (Zhao et al., 2022) proposed a hybrid Transformer–CNN AutoEncoder, where the Transformer encodes patch-level contextual features into a latent representation, and the CNN decodes it to reconstruct the clean image. Later, Phutke et al. (Phutke et al., 2023) integrated omni-dimensional gated attention with wavelet query modules to directly reconstruct clean images. Wang et al. (Wang et al., 2025) introduced an attention-driven forgery adversarial network (AFAN) that combines adversarial learning with forgery detection to guide end-to-end training.

While training an end-to-end network may seem straightforward, several studies instead formulate the **BI** task as a combination of mask prediction and image inpainting. Following this perspective, researchers have developed various **two-branch frameworks** that leverage the predicted mask information to guide the image reconstruction process. Wang et al. (Wang et al., 2020) introduced the Visual Consistency Network (VCN), a two-stage framework that first predicts the corrupted region mask and then reconstructs the clean image based on the predicted mask. Subsequently, Wang et al. (Wang et al., 2023b) incorporated frequency-domain information into the mask detection stage and employed a Top-Down Refinement Network for image reconstruction in the face inpainting domain. More recently, Zhao et al. (Zhao et al., 2025) proposed the Context-Aware Mutual Learning (CAML) framework, which jointly learns mask estimation and image inpainting through two interconnected branches that exchange contextual information.

Recently, the works related to image editing arised and have some capacity in solving **BI** problem. Commercial products like Flux (Labs et al., 2025), Nano Banana Pro (Labs et al., 2025)

A.2 Proof of Proposition 2.1

Proof. For a mask support $S = \mathrm{supp}(\boldsymbol{m})$, $\|(\mathbf{1} - \boldsymbol{m}) \odot \boldsymbol{r}\|_p^p = \sum_{i \notin S} r_i^p$. Removing one additional positive summand cannot increase the objective. An exchange argument then shows that if $i \in S$,

$j \notin S$, and $r_j > r_i$, replacing i by j strictly decreases the objective. Hence an optimal support contains the k largest residual magnitudes, which gives **??**. □

A.3 Proof of Proposition 2.2

Proof. For fixed $\boldsymbol{z}$, let $R_{ij} = (\boldsymbol{y} - G_{\boldsymbol{\theta}}(\boldsymbol{z}))_{ij}$. Under the squared ℓ_2 reconstruction loss, the optimization over $\boldsymbol{M}$ is separable across its entries. Hence, each M_{ij} is obtained by solving

$$\min_{m \in \mathbb{R}} \; R_{ij}^2(1-m)^2 + \lambda|m|.$$

For $m > 0$, the first-order optimality condition gives $2R_{ij}^2(m-1) + \lambda = 0$, yielding $m = 1 - \lambda/(2R_{ij}^2)$, which is positive if and only if $R_{ij}^2 > \lambda/2$. At $m = 0$, the subgradient condition is satisfied whenever $R_{ij}^2 \leq \lambda/2$. A negative minimizer cannot occur, since the corresponding stationary point lies in the positive domain.

Therefore,

$$M_{ij}^\star = \left(1 - \frac{\lambda}{2R_{ij}^2}\right)_+.$$

When $R_{ij}^2 > \lambda/2$, we have $0 < M_{ij}^\star < 1$ for any finite $\lambda > 0$. Thus, the ℓ_1-penalized formulation generally produces fractional active mask entries rather than enforcing $M_{ij}^\star \in \{0, 1\}$. Moreover, $M_{ij}^\star \to 1$ for any fixed nonzero residual only in the limit $\lambda \to 0^+$. □

A.4 Prompt Used for Conditional Image Generation

Identify any corrupted, missing, or occluded regions in the image without using an explicit mask. Restore those regions by plausibly reconstructing the original content, preserving the global structure, semantics, textures, and lighting of the image. Do not add new objects or alter intact regions.

A.5 Overfitting Issue for RGI

Proposition A.1 (Degeneracy under exact interpolation)**.** *Consider the mask-constrained formulation in Eq.* (4)*, and assume that the reconstruction loss is nonnegative and vanishes when its two arguments coincide. If the observation lies in the range of the generator, i.e., there exists some $\bar{\boldsymbol{z}}$ such that $G_{\boldsymbol{\theta}}(\bar{\boldsymbol{z}}) = \boldsymbol{y}$, then every feasible mask $\boldsymbol{M}$ with $\|\boldsymbol{M}\|_0 \leq k$ yields a global solution $(\boldsymbol{M}, \bar{\boldsymbol{z}})$. Consequently, the corruption mask is unidentifiable from the objective.*

Proof. Let $\boldsymbol{M}$ be any feasible mask satisfying $\|\boldsymbol{M}\|_0 \leq k$. By assumption, there exists $\bar{\boldsymbol{z}}$ such that

$$G_{\boldsymbol{\theta}}(\bar{\boldsymbol{z}}) = \boldsymbol{y}.$$

Therefore, after applying the mask in the reconstruction term, the two arguments of the reconstruction loss coincide. Hence, by the assumed property of the loss,

$$\ell((\mathbf{1} - \boldsymbol{M}) \odot \boldsymbol{y}, (\mathbf{1} - \boldsymbol{M}) \odot G_{\boldsymbol{\theta}}(\bar{\boldsymbol{z}})) = 0$$

where the notation indicates the mask-constrained reconstruction loss appearing in Eq. (4). Since the reconstruction loss is nonnegative, zero is the smallest possible objective value. Thus, $(\boldsymbol{M}, \bar{\boldsymbol{z}})$ is a global minimizer for every feasible $\boldsymbol{M}$ with $\|\boldsymbol{M}\|_0 \leq k$.
In particular, the objective assigns the same optimal value to all feasible masks, so it cannot distinguish the true corruption support from any other feasible support. Hence the corruption mask is unidentifiable from the objective. □

A.6 Detailed Experiment Setup for Anomaly Detection

Dataset Following RGI (Mou et al., 2023), we evaluated the proposed method on a synthetic defect dataset generated from Product03 of the BTAD dataset (Mishra et al., 2021). The dataset contains 900 defect-free images, which allows for the training of the DGPs. For evaluation, it provides five distinct defect scenarios: crack, scratch, irregular (irr.), mixed large (large), and spot. Each defect scenario contains 100 test samples.

Competeing Methods We compare our **ZSBI-FM** method with other state-of-the-art methods from both zero-shot and supervised learning paradigms. For the zero-shot setting, we consider AnoGAN (Schlegl et al., 2017) and RGI (Mou et al., 2023). To ensure a fair comparison among the zero-shot methods, we train a flow-matching backbone (Lipman et al., 2024) using the same 900 defect-free samples and adapt it to each method. For the supervised learning setting, we consider DRAEM (Zavrtanik et al., 2021) and Flux.2 Labs (2025). For the DRAEM method, we adapted their default setup and trained the model based on the simulated BTAD dataset, using DTD(Cimpoi et al., 2014) as the source of anomaly textures.

Infer Mask Region As the blind inpainting focuses more on the reconstruction, we directly look at the reconstructed image here and identify anomalous regions using the pixel-wise ℓ_1 residual between the reconstructed image and the input image, i.e., $\boldsymbol{r}_{ij} = \|\hat{\boldsymbol{x}}_{ij} - \boldsymbol{y}_{ij}\|_1$. Since the residual is evaluated as continuous-values and cannot directly serve as binary labels for anomalous regions, we applied a threshold selected using a small validation set. The thresholded scores were then used to generate the final binary anomaly map.

Metrics. Here we evaluate both the image reconstruction quality and the mask detection qualities. We used the same metric as we did in Sec. 4.1 to evaluate the reconstruction quality. We used the Dice coefficient, Average Precision (AP), and AUC-ROC to evaluate the pixel-wise anomaly detection performance. The dice coefficient is defined as

$$2 * \|\hat{\boldsymbol{M}} \odot \boldsymbol{M_{gt}}\|_0 / (\|\hat{\boldsymbol{M}}\|_0 + \|\boldsymbol{M_{gt}}\|_0) \tag{12}$$

Where both the predicted mask $\hat{\boldsymbol{M}}$ and the ground truth defective region $\boldsymbol{M_{gt}}$ are binary valued with 1 indicating the defective pixels. Note that for anormaly detection, the defect region will only cover a small region of the image, which introduced the class imbalance. Therefore directly comparing the results using AUC-ROC score is misleading. Here, we will mainly compare the Dice coefficient and AP among different methods.

A.7 Complete Results for FFHQ

Table 4: **Results for Brush & Noise and Brush & Solid Color corruptions on FFHQ**. Each corruption type is evaluated on 100 images. (**Bold**: best, under: second best, green: performance increase, red: performance decrease)

Method	Brush / Noise						Brush / Solid color					
	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	CLIP-IQA↑	MUSIQ↑	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	CLIP-IQA↑	MUSIQ↑
CAML	23.42	0.809	0.186	0.113	0.740	66.79	24.51	0.851	0.140	0.094	0.721	66.97
OmniNet	20.11	0.861	0.211	0.190	**0.798**	**70.44**	19.55	0.886	0.202	0.200	0.591	**72.32**
VCNet	21.51	0.824	0.214	0.178	0.687	66.49	17.46	0.730	0.308	0.286	0.645	64.73
FLUX.2-klein	25.15	0.870	0.098	0.090	0.737	66.89	25.07	0.873	0.103	0.095	0.743	66.24
FLUX.2-dev	28.94	0.913	**0.059**	0.052	0.753	67.59	29.61	0.917	**0.059**	0.050	**0.761**	67.78
RGI-DM	31.07	0.925	0.073	0.053	0.734	67.13	29.80	0.917	0.079	0.060	0.742	66.81
ZSBI-FM (Ours)	**32.30**	**0.936**	0.069	**0.050**	0.709	67.27	**32.88**	**0.942**	0.062	**0.047**	0.712	66.70
Ours vs. best	+1.23	+0.011	+0.010	−0.002	−0.088	−3.17	+3.08	+0.024	+0.003	−0.003	−0.049	−5.63

Table 5: **Results for Box & Noise and Box & Solid Color corruptions on FFHQ**. Each corruption type is evaluated on 100 images. (**Bold**: best, under: second best, green: performance increase, red: performance decrease)

Method	Box / Noise						Box / Solid color					
	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	CLIP-IQA↑	MUSIQ↑	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	CLIP-IQA↑	MUSIQ↑
CAML	20.78	0.756	0.251	0.177	0.700	**69.55**	19.27	0.774	0.252	0.175	0.541	67.42
OmniNet	20.28	0.901	0.126	0.148	0.552	69.43	20.40	0.929	0.122	0.129	0.413	**69.54**
VCNet	19.89	0.809	0.200	0.199	0.511	64.96	18.49	0.787	0.234	0.222	0.420	64.43
FLUX.2-klein	24.55	0.867	0.094	0.086	0.747	67.49	25.66	0.890	0.078	0.074	0.758	67.96
FLUX.2-dev	27.96	0.916	**0.054**	0.049	**0.755**	67.93	28.52	0.919	**0.052**	0.048	**0.759**	67.56
RGI-DM	28.93	0.921	0.061	0.050	0.734	67.37	28.62	0.917	0.061	0.056	0.738	67.72
ZSBI-FM (Ours)	**30.88**	**0.936**	0.056	**0.045**	0.719	67.38	**31.36**	**0.939**	0.052	**0.044**	0.717	66.84
Ours vs. best	+1.95	+0.016	+0.002	−0.004	−0.036	−2.17	+2.74	+0.010	+0.001	−0.004	−0.042	−2.70

Table 6: **Results for Patches & Noise and Patches & Solid Color corruptions on FFHQ**. Each corruption type is evaluated on 100 images. (**Bold**: best, under: second best, green: performance increase, red: performance decrease)

Method	Patches / Noise						Patches / Solid color					
	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	CLIP-IQA↑	MUSIQ↑	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	CLIP-IQA↑	MUSIQ↑
CAML	23.64	0.810	0.184	0.116	0.757	67.62	24.00	0.839	0.153	0.099	0.731	67.40
OmniNet	20.81	0.892	0.163	0.166	0.766	**70.19**	20.20	0.910	0.159	0.173	0.620	**71.30**
VCNet	25.95	0.890	0.131	0.112	0.684	66.49	17.10	0.712	0.300	0.293	0.648	61.57
FLUX.2-klein	25.65	0.887	0.089	0.083	0.740	67.10	25.96	0.893	0.081	0.074	0.749	67.30
FLUX.2-dev	28.80	0.914	**0.057**	0.050	**0.767**	67.75	29.35	0.924	**0.052**	0.048	**0.759**	67.86
RGI-DM	30.71	0.928	0.065	0.049	0.729	67.05	29.79	0.923	0.064	0.053	0.741	67.23
ZSBI-FM (Ours)	**32.22**	**0.940**	0.061	**0.047**	0.715	67.37	**32.36**	**0.942**	0.056	**0.045**	0.713	67.12
Ours vs. best	+1.50	+0.012	+0.005	−0.003	−0.052	−2.82	+2.57	+0.018	+0.004	−0.003	−0.045	−4.19

Table 7: **Results for Text and Brush & DTD corruptions on FFHQ**. Each corruption type is evaluated on 100 images. (**Bold**: best, under: second best, green: performance increase, red: performance decrease)

Method	Text						Brush / DTD					
	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	CLIP-IQA↑	MUSIQ↑	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	CLIP-IQA↑	MUSIQ↑
CAML	23.06	0.802	0.192	0.124	0.731	68.15	23.85	0.836	0.153	0.103	0.743	66.98
OmniNet	22.24	0.920	0.123	0.125	**0.782**	**70.89**	20.04	0.876	0.194	0.189	**0.774**	**70.93**
VCNet	13.07	0.406	0.481	0.454	0.498	43.08	14.94	0.553	0.404	0.384	0.589	52.21
FLUX.2-klein	25.05	0.882	0.094	0.092	0.748	66.82	25.33	0.881	0.092	0.086	0.751	67.13
FLUX.2-dev	29.10	0.924	**0.055**	0.052	0.761	67.90	29.36	0.917	**0.057**	**0.051**	0.764	67.80
RGI-DM	31.28	0.933	0.065	0.052	0.727	66.29	28.78	0.909	0.081	0.064	0.751	66.48
ZSBI-FM (Ours)	**33.30**	**0.947**	0.056	**0.042**	0.721	66.95	**31.32**	**0.931**	0.067	0.054	0.721	66.08
Ours vs. best	+2.02	+0.014	+0.001	−0.010	−0.062	−3.94	+1.96	+0.014	+0.010	+0.003	−0.054	−4.85

Table 8: **Results for Box & DTD and Patches & DTD corruptions on FFHQ**. Each corruption type is evaluated on 100 images. (**Bold**: best, under: second best, green: performance increase, red: performance decrease)

Method	Box / DTD						Patches / DTD					
	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	CLIP-IQA↑	MUSIQ↑	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	CLIP-IQA↑	MUSIQ↑
CAML	19.74	0.763	0.244	0.175	0.667	68.73	23.99	0.842	0.146	0.098	0.734	67.51
OmniNet	21.04	0.916	0.112	0.130	0.545	**69.18**	20.71	0.899	0.154	0.161	0.712	**70.50**
VCNet	17.19	0.696	0.284	0.271	0.521	59.25	14.71	0.534	0.405	0.389	0.567	50.63
FLUX.2-klein	25.47	0.881	0.085	0.077	0.756	67.87	25.54	0.882	0.087	0.084	0.752	67.49
FLUX.2-dev	28.69	0.919	**0.052**	0.048	**0.757**	67.53	29.12	0.924	**0.052**	0.049	**0.759**	67.86
RGI-DM	28.01	0.914	0.063	0.054	0.735	67.45	29.04	0.917	0.067	0.058	0.736	66.84
ZSBI-FM (Ours)	**30.78**	**0.934**	0.054	**0.043**	0.718	66.55	**31.78**	**0.938**	0.057	**0.046**	0.723	66.19
Ours vs. best	+2.10	+0.015	+0.002	−0.006	−0.040	−2.63	+2.66	+0.014	+0.005	−0.002	−0.036	−4.31

A.8 Complete Results for ImageNet

Table 9: **Results for Brush & Noise and Brush & Solid Color corruptions on ImageNet**. Each corruption type is evaluated on 100 images. (**Bold**: best, under: second best, green: performance increase, red: performance decrease)

Method	Brush / Noise						Brush / Solid color					
	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	CLIP-IQA↑	MUSIQ↑	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	CLIP-IQA↑	MUSIQ↑
CAML	18.82	0.653	0.365	0.214	0.744	53.50	19.09	0.688	0.331	0.206	0.656	50.46
FLUX.2-klein	22.45	0.802	0.148	0.119	0.830	59.75	22.16	0.797	0.154	0.130	0.826	59.05
FLUX.2-dev	25.07	0.851	**0.100**	**0.080**	**0.859**	59.59	25.55	0.863	**0.090**	**0.075**	**0.851**	59.52
RGI-DM	25.97	0.858	0.122	0.094	0.847	60.18	25.87	0.854	0.122	0.092	0.829	58.22
ZSBI-FM (Ours)	**26.72**	**0.875**	0.109	0.090	0.853	**61.36**	**27.15**	**0.884**	0.103	0.085	0.834	**60.67**
Ours vs. best	+0.76	+0.017	+0.009	+0.010	−0.006	+1.17	+1.28	+0.021	+0.013	+0.009	−0.016	+1.15

Table 10: **Results for Box & Noise and Box & Solid Color corruptions on ImageNet**. Each corruption type is evaluated on 100 images. (**Bold**: best, under: second best, green: performance increase, red: performance decrease)

Method	Box / Noise						Box / Solid color					
	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	CLIP-IQA↑	MUSIQ↑	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	CLIP-IQA↑	MUSIQ↑
CAML	17.81	0.636	0.377	0.226	0.704	55.71	16.96	0.649	0.389	0.235	0.449	52.67
FLUX.2-klein	21.92	0.808	0.144	0.118	0.819	59.47	22.65	0.834	0.116	0.100	0.842	59.93
FLUX.2-dev	24.18	0.854	**0.096**	**0.079**	**0.846**	59.47	24.55	0.865	**0.091**	**0.076**	**0.845**	59.66
RGI-DM	24.88	0.860	0.107	0.089	0.825	59.35	24.56	0.860	0.105	0.087	0.825	59.43
ZSBI-FM (Ours)	**26.07**	**0.879**	0.100	0.084	0.838	**60.83**	**26.33**	**0.884**	0.092	0.078	0.842	**60.28**
Ours vs. best	+1.19	+0.020	+0.004	+0.005	−0.008	+1.36	+1.77	+0.018	+0.001	+0.001	−0.004	+0.34

Table 11: **Results for Patches & Noise and Patches & Solid Color corruptions on ImageNet**. Each corruption type is evaluated on 100 images. (**Bold**: best, under: second best, green: performance increase, red: performance decrease)

Method	Patches / Noise						Patches / Solid color					
	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	CLIP-IQA↑	MUSIQ↑	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	CLIP-IQA↑	MUSIQ↑
CAML	18.77	0.655	0.364	0.216	0.723	53.30	18.62	0.673	0.346	0.215	0.649	50.29
FLUX.2-klein	22.23	0.809	0.141	0.120	0.837	60.79	22.87	0.819	0.133	0.112	0.822	59.47
FLUX.2-dev	24.90	0.854	**0.097**	**0.082**	**0.845**	59.87	25.20	0.853	0.096	**0.077**	**0.844**	59.28
RGI-DM	25.47	0.862	0.112	0.093	0.830	59.74	25.53	0.862	0.104	0.084	0.837	58.87
ZSBI-FM (Ours)	**26.26**	**0.879**	0.104	0.090	0.842	**61.36**	**27.15**	**0.885**	**0.096**	0.082	0.832	**60.25**
Ours vs. best	+0.79	+0.017	+0.008	+0.008	−0.003	+0.57	+1.61	+0.023	−0.001	+0.005	−0.012	+0.78

Table 12: **Results for Text and Brush & DTD corruptions on ImageNet**. Each corruption type is evaluated on 100 images. (**Bold**: best, under: second best, green: performance increase, red: performance decrease)

Method	Text						Brush / DTD					
	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	CLIP-IQA↑	MUSIQ↑	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	CLIP-IQA↑	MUSIQ↑
CAML	18.95	0.659	0.359	0.216	0.665	52.06	18.92	0.667	0.345	0.212	0.680	51.50
FLUX.2-klein	22.44	0.816	0.145	0.119	0.828	59.56	21.46	0.787	0.172	0.143	0.836	**60.69**
FLUX.2-dev	25.94	0.873	**0.086**	**0.075**	**0.852**	59.23	**25.34**	0.858	**0.098**	**0.082**	**0.862**	59.88
RGI-DM	26.87	0.880	0.107	0.085	0.829	59.55	24.31	0.845	0.132	0.109	0.835	59.20
ZSBI-FM (Ours)	**27.65**	**0.893**	0.095	0.082	0.841	**60.46**	25.13	**0.860**	0.122	0.099	0.853	60.49
Ours vs. best	+0.78	+0.014	+0.009	+0.006	−0.011	+0.90	−0.22	+0.002	+0.024	+0.017	−0.009	−0.20

Table 13: **Results for Box & DTD and Patches & DTD corruptions on ImageNet**. Each corruption type is evaluated on 100 images. (**Bold**: best, under: second best, green: performance increase, red: performance decrease)

Method	Box / DTD						Patches / DTD					
	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	CLIP-IQA↑	MUSIQ↑	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	CLIP-IQA↑	MUSIQ↑
CAML	17.61	0.647	0.365	0.225	0.570	53.75	18.44	0.662	0.345	0.214	0.655	51.58
FLUX.2-klein	21.88	0.811	0.141	0.119	0.813	**60.45**	21.37	0.804	0.155	0.131	0.838	**60.63**
FLUX.2-dev	23.83	0.854	**0.099**	**0.085**	**0.843**	59.69	**25.03**	0.861	**0.093**	**0.079**	**0.856**	60.08
RGI-DM	23.80	0.855	0.112	0.094	0.821	59.78	23.83	0.849	0.127	0.104	0.829	59.57
ZSBI-FM (Ours)	**25.04**	**0.870**	0.101	0.085	0.821	59.83	24.68	**0.867**	0.114	0.096	0.848	60.50
Ours vs. best	+1.21	+0.015	+0.001	+0.0002	−0.023	−0.62	−0.36	+0.007	+0.021	+0.017	−0.007	−0.13

A.9 Complete Results for Simulated BTAD Dataset

Table 14: **Crack anomaly type on the simulated BTAD3 dataset**. Each metric is averaged over 100 test images. (**Bold**: best; under: second best; green: improvement; red: degradation.)

Method	Reconstruction Quality						Residual-Mask Quality		
	PSNR↑	SSIM↑	LPIPS↓	CLIP-IQA↑	DISTS↓	MUSIQ↑	ROC-AUC↑	AP↑	Dice↑
Flux.2	19.39	0.764	0.124	0.653	0.128	33.46	0.696	0.083	0.118
AnoGan	19.51	0.697	0.136	**0.814**	0.093	**42.87**	0.922	0.151	0.248
RGI	19.21	0.703	0.137	0.745	0.101	39.49	0.915	0.132	0.230
DRAEM	27.28	0.881	0.103	0.717	0.091	40.94	0.993	0.807	0.670
ZSBI-FM (Ours)	**29.31**	**0.916**	**0.067**	0.801	**0.051**	42.32	**0.999**	**0.931**	**0.888**
Ours vs. best	+2.04	+0.035	−0.036	−0.013	−0.040	−0.56	+0.006	+0.125	+0.218

Table 15: **Scratch anomaly type on the simulated BTAD3 dataset**. Each metric is averaged over 100 test images. (**Bold**: best; under: second best; green: improvement; red: degradation.)

Method	Reconstruction Quality						Residual-Mask Quality		
	PSNR↑	SSIM↑	LPIPS↓	CLIP-IQA↑	DISTS↓	MUSIQ↑	ROC-AUC↑	AP↑	Dice↑
Flux.2	19.23	0.756	0.121	0.648	0.128	32.98	0.631	0.044	0.063
AnoGan	19.47	0.696	0.135	**0.810**	0.092	**42.71**	0.912	0.143	0.229
RGI	19.22	0.703	0.137	0.737	0.102	39.47	0.906	0.122	0.208
DRAEM	27.29	0.881	0.103	0.718	0.091	40.94	0.992	0.752	0.628
ZSBI-FM (Ours)	**29.26**	**0.916**	**0.067**	0.803	**0.051**	42.27	**0.998**	**0.896**	**0.857**
Ours vs. best	+1.97	+0.035	−0.036	−0.006	−0.040	−0.43	+0.006	+0.144	+0.228

Table 16: **Irregular anomaly type on the simulated BTAD3 dataset**. Each metric is averaged over 100 test images. (**Bold**: best; under: second best; green: improvement; red: degradation.)

Method	Reconstruction Quality						Residual-Mask Quality		
	PSNR↑	SSIM↑	LPIPS↓	CLIP-IQA↑	DISTS↓	MUSIQ↑	ROC-AUC↑	AP↑	Dice↑
Flux.2	19.19	0.757	0.117	0.665	0.127	33.47	0.675	0.067	0.089
AnoGan	19.55	0.699	0.134	**0.809**	0.092	**42.83**	0.916	0.120	0.191
RGI	19.22	0.703	0.137	0.743	0.101	39.59	0.907	0.108	0.176
DRAEM	27.28	0.881	0.103	0.717	0.091	40.95	0.996	0.760	0.599
ZSBI-FM (Ours)	**29.26**	**0.916**	**0.067**	0.801	**0.051**	42.31	**0.999**	**0.904**	**0.859**
Ours vs. best	+1.98	+0.035	−0.036	−0.008	−0.040	−0.52	+0.003	+0.144	+0.260

Table 17: **Large anomaly type on the simulated BTAD3 dataset**. Each metric is averaged over 100 test images. (**Bold**: best; under: second best; green: improvement; red: degradation.)

Method	Reconstruction Quality						Residual-Mask Quality		
	PSNR↑	SSIM↑	LPIPS↓	CLIP-IQA↑	DISTS↓	MUSIQ↑	ROC-AUC↑	AP↑	Dice↑
Flux.2	18.50	0.733	0.143	0.602	0.141	33.45	0.732	0.272	0.288
AnoGan	19.45	0.694	0.136	**0.817**	0.093	**42.59**	0.915	0.377	0.363
RGI	18.52	0.676	0.155	0.736	0.116	37.87	0.840	0.279	0.290
DRAEM	26.75	0.875	0.106	0.711	0.093	40.74	0.995	0.921	0.849
ZSBI-FM (Ours)	**27.67**	**0.900**	**0.073**	0.798	**0.054**	42.24	**0.998**	**0.962**	**0.935**
Ours vs. best	+0.92	+0.025	−0.033	−0.019	−0.039	−0.35	+0.003	+0.041	+0.086

Table 18: **Spot anomaly type on the simulated BTAD3 dataset**. Each metric is averaged over 100 test images. (**Bold**: best; under: second best; green: improvement; red: degradation.)

Method	Reconstruction Quality						Residual-Mask Quality		
	PSNR↑	SSIM↑	LPIPS↓	CLIP-IQA↑	DISTS↓	MUSIQ↑	ROC-AUC↑	AP↑	Dice↑
Flux.2	18.68	0.741	0.125	0.644	0.131	33.17	0.638	0.054	0.064
AnoGan	19.57	0.698	0.135	**0.808**	0.092	**42.61**	0.905	0.127	0.173
RGI	19.24	0.704	0.137	0.746	0.102	39.33	0.894	0.107	0.157
DRAEM	27.24	0.881	0.103	0.719	0.091	40.94	0.991	0.687	0.544
ZSBI-FM (Ours)	**29.15**	**0.915**	**0.067**	0.801	**0.051**	42.29	**0.998**	**0.892**	**0.844**
Ours vs. best	+1.90	+0.035	−0.036	−0.008	−0.040	−0.32	+0.007	+0.205	+0.300