LNRR: LEARNABLE NONLOCAL RELATION REGULARIZATION FOR IMPLICIT NEURAL REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Implicit neural representations (INRs) model discrete observations as samples of a continuous signal and provide compact, resolution-independent representations. However, an INR is supervised only at observed locations and generalizes poorly to unobserved ones, especially inside large missing regions. Local smoothness priors only propagate information inward from the boundary of a missing region, so the interior of a large region degrades into a blur that loses the original edges and textures. These structures can only be borrowed from similar ones elsewhere in the image, yet INRs have no built-in mechanism to exploit such nonlocal relations. We propose Learnable Nonlocal Relation Regularization (LNRR), which brings nonlocal relations between image patches into INR training to improve completion under large missing regions. LNRR learns patch relations with a patch feature network and uses them to constrain the reconstruction of missing regions through a robust graph-residual regularizer in the wavelet domain. LNRR changes only the regularization term and applies to a range of INR backbones. Across multiple natural-image datasets and missing patterns, LNRR achieves state-of-the-art completion among existing INR regularization methods.

1 INTRODUCTION

Implicit neural representations (INRs) have recently become an emerging and popular self-supervised approach to the compressed representation of signals (Dupont et al., 2021; Chen et al., 2021; Xie et al., 2022; Ni et al., 2026). An INR parameterizes a signal as a continuous function with a coordinate network that is fitted to the signal itself (Sitzmann et al., 2020; Tancik et al., 2020). INRs have been successfully applied in many areas, such as novel view synthesis (Mildenhall et al., 2020), 3D shape modeling (Park et al., 2019; Mescheder et al., 2019), and scientific data compression (Lu et al., 2021; Huang & Hoefler, 2023; Ni et al., 2026). Early INRs suffered from a spectral bias toward low frequencies (Rahaman et al., 2019). Sinusoidal and wavelet activations (Sitzmann et al., 2020; Saragadam et al., 2023; Liu et al., 2024), Fourier features and hash-grid encodings (Tancik et al., 2020; Müller et al., 2022), and multiplicative filter architectures (Fathony et al., 2021) have since greatly improved their ability to represent high frequencies, and fitting accuracy under complete observations is no longer the bottleneck. When observations are incomplete, the bottleneck shifts to generalization. Here, generalization refers to prediction from observed to unobserved coordinates within a single signal. The network is supervised only at observed locations, and its values at missing locations are determined entirely by its inductive bias. Designs that strengthen high-frequency expressiveness weaken the low-frequency bias that otherwise acts as an implicit regularizer, and predictions at missing locations degrade accordingly. Li et al. (2023) observe that the frequency hyperparameter giving the best INR generalization changes with the missing rate.

The problem is most severe when missing pixels form large contiguous regions. The interior of such a region is far from any observation and receives no supervision. The low-frequency bias of the INR and common regularizers such as total variation (TV) (Rudin et al., 1992) favor similar values at neighboring locations, so they only propagate information inward from the boundary. The larger the region, the more its interior degrades into a blur without the original edges and textures.

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.

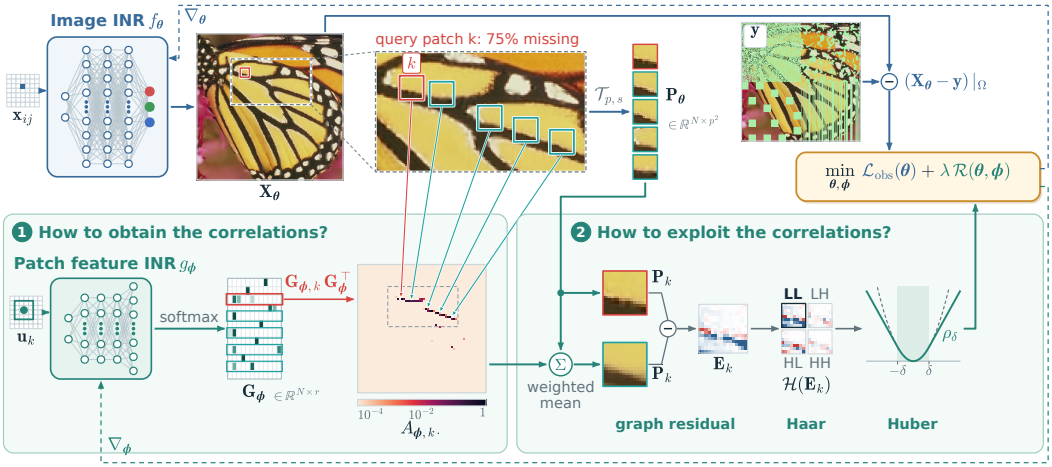

Figure 1: **Overview of LNRR.** The blue path is a plain INR, which renders the image from pixel coordinates and fits it to the observed pixels. The regularizer of LNRR comprises the patch extraction $\mathcal{T}_{p,s}$ in the middle and the two lower panels. In panel 1, a patch feature network maps each patch center to a feature, and the inner products of these features give the relations between patches (Section 3.2). In panel 2, a Huber potential on the Haar coefficients of the graph residual pulls each patch toward the weighted mean of its related patches (Section 3.3). The two networks are trained jointly. The red box marks an example patch, and the cyan boxes mark its four most related neighbors.

Priors trained on external data require a training set drawn from the same distribution as the target, whereas an INR is usually fitted to a single signal on its own. Without external data, the content that belongs inside a missing region can only be borrowed from elsewhere in the image. In natural images, edges, corners, and textures recur at different locations, so a missing structure often has similar instances elsewhere. This is nonlocal self-similarity, the basis of classical methods such as nonlocal means and BM3D (Buades et al., 2005; Dabov et al., 2007). INRs have no built-in mechanism to exploit it, since the network only ties nearby coordinates together. INRR (Li et al., 2023) introduces nonlocal graphs between rows and between columns into INR training. However, it still struggles to relate two similar but distant regions, and recovering structure inside large missing regions further requires capturing nonlocal relations between patches. Borrowing similar structures from elsewhere raises two questions.

How can the relations be obtained? Deciding whether two patches are similar normally requires comparing their contents, which are unknown inside missing regions. Classical nonlocal methods rely on pixel distances between patches, which are unavailable here. INRR learns relations with a small INR and avoids pixel comparisons, but its nodes are entire rows and columns. Recurring structures are 2D and local, whereas whole rows and columns containing them are usually dissimilar.

How can the relations be exploited? Similar patches still differ in brightness and detail, and learned relations are never entirely correct. Requiring connected patches to be close is therefore too rigid, because it smooths away genuine details and carries wrong relations into the reconstruction.

We propose Learnable Nonlocal Relation Regularization (LNRR) to answer both questions (Figure 1).

Obtaining the relations. LNRR takes overlapping 2D image patches as nodes. A patch feature network maps the center coordinate of each patch to a patch feature, and inner products between features give the relations between patches. Patches inside a missing region also have features and can be connected to patches anywhere in the image. The patch feature network and the image INR are trained jointly from random initialization. Observed information enters relation learning through the reconstruction, and the learned relations in turn constrain the reconstruction at missing locations.

Exploiting the relations. LNRR turns the learned relations into a constraint on the reconstruction in three steps. The first step computes the graph residual, the deviation of each patch from the weighted mean of its similar patches. A patch thus only needs to be close to this mean rather than to each similar patch. The second step applies the Huber potential to the residual. The potential is quadratic for small

deviations and bounds the gradient of large ones, so wrong relations cannot dominate training. The third step evaluates the potential on the Haar wavelet coefficients of the residual. Uniform brightness offsets between similar patches enter only the low-frequency coefficients and are relaxed there, while the detail coefficients carrying edges and textures remain fully constrained.

LNRR replaces only the regularization term of the training objective and applies to different INR backbones. It requires no external data, and the patch feature network is discarded at inference.

Our contributions are as follows:

1. We take overlapping 2D image patches as nodes and learn inter-patch relations from coordinates with a patch feature network trained jointly with the image INR, so that patches inside missing regions can connect to similar structures elsewhere.
2. We design a robust wavelet-domain graph-residual energy that turns the learned relations into a constraint on the reconstruction. It tolerates differences between similar patches as well as wrong relations and thus preserves edges and textures in missing regions.
3. We evaluate LNRR on multiple datasets under five missing patterns. With a SIREN backbone, LNRR achieves better completion than CRNL, INRR, and TV on every missing pattern. On the most challenging patch mask, it also gives the highest mean PSNR with four other INR backbones. A mechanism analysis shows that the learned relations connect missing patches to similar structures elsewhere in the image.

2 Related Work

Generalization and regularization of INRs. To counter the spectral bias (Rahaman et al., 2019), prior work strengthens the high-frequency representation of INRs through activation functions (Sitzmann et al., 2020; Saragadam et al., 2023; Liu et al., 2024), coordinate encodings (Tancik et al., 2020; Müller et al., 2022), and network architectures (Fathony et al., 2021; Lindell et al., 2022; Shekarforoush et al., 2022), mostly under complete observations. With incomplete observations, the values at missing locations are determined by the inductive bias. Deep Image Prior uses the network architecture as a prior for single-image restoration (Ulyanov et al., 2018), RHINO adds a continuous input to hash encodings to restore regular interpolation (Zhu et al., 2026), and TV is the most common explicit regularizer (Rudin et al., 1992). All of these priors act locally. LRTFR and CP-Pruner impose low-rank and smoothness priors through low-rank functional decompositions (Luo et al., 2024; Cheng et al., 2026), mainly for multi-dimensional data completion under random sampling. Cross-instance generalization via hypernetworks or pretrained priors is a separate problem.

Nonlocal self-similarity. Nonlocal means, BM3D, and WNNM restore an image from similar patches within it (Buades et al., 2005; Dabov et al., 2007; Gu et al., 2014). As variational regularizers, nonlocal operators and functionals (Gilboa & Osher, 2008) have been applied to inverse problems with alternating updates of the graph and the image (Peyré et al., 2008) and to exemplar-based inpainting (Arias et al., 2011), and LDMM and CURE penalize the dimension and the curvature of the patch manifold (Osher et al., 2017; Dong et al., 2020). These methods compute relations from pixel distances, which are unreliable when patches contain missing pixels. With continuous representations, Neural Knitworks enforces consistency between patches predicted by an MLP (Czerkawski et al., 2024), CRNL hard-groups patches by Euclidean distance on a coarse reconstruction and applies coupled low-rank functional decompositions to the fixed groups (Luo et al., 2025), and TenF-INR adopts a similar idea for dynamic MRI (Liu et al., 2026). AIR and INRR learn the relations instead, parameterizing the Laplacian in matrix completion (Li et al., 2022) or row and column adjacencies with a small INR trained jointly with the image INR (Li et al., 2023). LNRR also generates relations with a coordinate network, but over overlapping 2D patches, and its soft adjacency evolves with the image without coarse reconstruction or patch matching.

Energies for graph regularization. Given the adjacency, the energy determines the effect of the regularizer. Quadratic energies, such as the nonlocal H^1 and graph Laplacian regularizers (Gilboa & Osher, 2008; Pang & Cheung, 2017) and the nonlocal weighted-mean residual of ASDS (Dong et al., 2011), force dissimilar connected nodes together, and nonlocal TV and p-Dirichlet energies use non-quadratic penalties instead (Kindermann et al., 2005; Gilboa & Osher, 2008; Elmoataz et al., 2008). Transform-domain methods constrain groups of similar patches by 3D shrinkage (Dabov et al., 2007), sparse dictionaries (Zhang et al., 2014), or tight frames (Cai et al., 2008), and NCSR, the closest to

LNRR, uses an ℓ_1 norm to penalize the deviation of each sparse code from an estimate formed from similar patches (Dong et al., 2013). All of these methods derive their weights or groups from pixel distances, possibly on preprocessed observations (Lou et al., 2010). Robust potentials downweight large residuals (Black & Rangarajan, 1996). LNRR applies the convex Huber potential with bounded influence (Huber, 1964) to the Haar coefficients of the graph residual (Section 3.3), which differs from the use of wavelets as INR activations (Saragadam et al., 2023) or signal representations (Yu et al., 2026).

3 METHOD

In this section, we design a new regularizer that learns and exploits the similarity between image patches. Section 3.1 gives the completion objective and the general form of nonlocal graph regularization. Section 3.2 obtains the relations between patches by learning a patch graph (Figure 1, lower left). Section 3.3 exploits these relations with a robust wavelet-domain graph-residual energy (Figure 1, lower right). Section 3.4 describes training and inference.

3.1 COMPLETION OBJECTIVE AND NONLOCAL GRAPH REGULARIZATION

A coordinate network $f_{\boldsymbol{\theta}} : \mathbb{R}^2 \to \mathbb{R}^C$ maps normalized pixel coordinates $\mathbf{x}_{ij} \in [-1, 1]^2$ to pixel values (Figure 1, top left). Backbones such as SIREN, FFN, WIRE, InstantNGP, and MFN differ in their coordinate encodings and activation functions (Appendix B.2). Let $\mathcal{G}$ be the image grid, $\Omega \subseteq \mathcal{G}$ the set of observed locations, $\mathbf{y}_{ij} \in \mathbb{R}^C$ the observed value at $(i, j) \in \Omega$, and $\mathbf{X}_{\boldsymbol{\theta}}[i, j, :] = f_{\boldsymbol{\theta}}(\mathbf{x}_{ij})$ the reconstruction. The regularized completion objective (Figure 1, top right) is

$$\min_{\boldsymbol{\theta}} \; \mathcal{L}_{\text{obs}}(\boldsymbol{\theta}) + \lambda \mathcal{R}(\mathbf{X}_{\boldsymbol{\theta}}), \qquad \mathcal{L}_{\text{obs}}(\boldsymbol{\theta}) = \frac{1}{C|\Omega|} \sum_{(i,j)\in\Omega} \| f_{\boldsymbol{\theta}}(\mathbf{x}_{ij}) - \mathbf{y}_{ij} \|_2^2 . \tag{1}$$

Missing locations receive gradients only through $\mathcal{R}$, and $\mathcal{R}$ acts only on the rendered pixel grid, independently of the backbone architecture. Nonlocal graph regularization (Gilboa & Osher, 2008; Peyré et al., 2008) builds a weighted graph over nodes and penalizes differences between connected nodes, typically through the Dirichlet energy

$$\mathcal{R}_{\text{DE}}(\mathbf{V}; \mathbf{A}) = \operatorname{tr}\big(\mathbf{V}^\top \mathbf{L} \mathbf{V}\big) = \frac{1}{2} \sum_{i,j} A_{ij} \, \| \mathbf{V}_i - \mathbf{V}_j \|_2^2 , \qquad \mathbf{L} = \operatorname{diag}(\mathbf{A}\mathbf{1}) - \mathbf{A}, \tag{2}$$

where $\mathbf{V}_i$ is the signal on node i and $\mathbf{A}$ is the adjacency. A graph regularizer thus has two components. The adjacency $\mathbf{A}$ specifies which nodes are related, and the energy specifies how related nodes constrain each other. These two components correspond to the two questions raised in Section 1.

Where does A come from? When $\mathbf{A}$ is computed from pixel distances between patches, the distances involving patches inside a missing region can only be computed from the current estimate, which is precisely the unknown. INRR (Li et al., 2023) avoids this by parameterizing row and column adjacencies with an auxiliary INR, but its nodes $\mathbf{V}_i$ are entire rows and columns, and two rows are rarely similar as a whole. Section 3.2 learns $\mathbf{A}$ with 2D patches as nodes.

Which energy should constrain connected nodes? Eq. (2) penalizes every edge difference quadratically and requires every connected pair to be close. Similar patches still differ, and the learned $\mathbf{A}$ is imperfect. A quadratic penalty therefore smooths away genuine differences, and wrong edges force dissimilar content together. Section 3.3 instead constructs a graph-residual energy.

3.2 PATCH GRAPH: OBTAINING INTER-PATCH RELATIONS

Nodes. The nodes are overlapping image patches. For channel c of the reconstruction, the operator $\mathcal{T}_{p,s}$ in Figure 1 extracts patches of width p with stride s:

$$\mathbf{P}_{\boldsymbol{\theta}}^{(c)} = \mathcal{T}_{p,s}\Big(\mathbf{X}_{\boldsymbol{\theta}}^{(c)}\Big) \in \mathbb{R}^{N \times p^2}, \qquad N = \left\lceil \frac{m}{s} \right\rceil \left\lceil \frac{n}{s} \right\rceil , \tag{3}$$

where $m \times n$ is the image size and row k of the linear operator $\mathcal{T}_{p,s}$ outputs a flattened $p \times p$ patch, with missing locations filled by the network prediction. The patch scale is comparable to that of edge and texture elements. Patch centers are placed every s pixels (Figure 1, middle; Appendix A.1). With

$p = 16$ and $s = 4$, each interior pixel falls into 16 patches. Relative positions within a patch are fixed, so any two patches have a pixel-wise correspondence.

Edges. Edges between patches cannot be computed from pixel distances (Section 3.1). We follow INRR in parameterizing relations with a coordinate network, but we take 2D patch center coordinates as input (Figure 1, lower left). The patch feature network $g_{\boldsymbol{\phi}} : \mathbb{R}^2 \to \mathbb{R}^r$ is a SIREN with parameters independent of $f_{\boldsymbol{\theta}}$. It maps the normalized center coordinate $\mathbf{u}_k \in [-1, 1]^2$ of patch k to a patch feature, and inner products between features give the adjacency:

$$\mathbf{G}_{\boldsymbol{\phi},k} = \mathrm{softmax}(g_{\boldsymbol{\phi}}(\mathbf{u}_k)) \in \mathbb{R}^r, \qquad A_{\boldsymbol{\phi},kj} = \begin{cases} \langle \mathbf{G}_{\boldsymbol{\phi},k}, \mathbf{G}_{\boldsymbol{\phi},j} \rangle, & k \neq j, \\ 0, & k = j. \end{cases} \tag{4}$$

The softmax makes $\mathbf{G}_{\boldsymbol{\phi},k}$ a soft assignment of patch k over r latent classes, so $A_{\boldsymbol{\phi},kj} \in [0, 1]$ is the probability that patches k and j fall into the same class when each draws a latent class independently. Any two patches can be connected, so recurring structures far apart can constrain each other (see the zoom-in of Figure 1). All channels share the same graph, and patches inside missing regions also receive features from $g_{\boldsymbol{\phi}}$.

Joint learning of the graph and the image. Both $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ start from random initialization and are updated jointly. Observed information enters the adjacency through the reconstructed patches, and the learned adjacency in turn constrains the predictions at missing locations. This adjacency is never entirely correct. Early in training, it is nearly uniform and connects unrelated patches. Later, connected patches often agree only at some positions and differ elsewhere because of misaligned edges, occlusions, or different content. The energy of Section 3.3 takes this into account when exploiting the relations.

3.3 Graph-Residual Energy: Exploiting Inter-Patch Relations

Given the patch graph, the energy turns the inter-patch relations into a constraint on the reconstruction. We build it in three steps, which decide what to penalize (the graph residual), how to penalize it (a robust potential), and in which basis (a transform domain). All three steps account for the fact that connected patches are similar rather than identical and that the learned relations are imperfect.

Graph residual. We penalize the deviation of each patch from the weighted mean of its neighbors:

$$\mathbf{E}^{(c)}_{\boldsymbol{\theta},\boldsymbol{\phi}} = \mathbf{L}_{\boldsymbol{\phi}} \mathbf{P}^{(c)}_{\boldsymbol{\theta}}, \qquad \mathbf{E}^{(c)}_{\boldsymbol{\theta},\boldsymbol{\phi},k} = \sum_j A_{\boldsymbol{\phi},kj} \left(\mathbf{P}^{(c)}_{\boldsymbol{\theta},k} - \mathbf{P}^{(c)}_{\boldsymbol{\theta},j} \right) = d_{\boldsymbol{\phi},k} \left(\mathbf{P}^{(c)}_{\boldsymbol{\theta},k} - \bar{\mathbf{P}}^{(c)}_{\boldsymbol{\theta},\boldsymbol{\phi},k} \right), \tag{5}$$

where $d_{\boldsymbol{\phi},k} = \sum_j A_{\boldsymbol{\phi},kj}$ is the degree of node k and $\bar{\mathbf{P}}^{(c)}_{\boldsymbol{\theta},\boldsymbol{\phi},k}$ is the adjacency-weighted mean of the neighbors of patch k ($\mathbf{E}_k$ and $\bar{\mathbf{P}}_k$ in Figure 1). Eq. (2) penalizes each edge difference separately, whereas the graph residual first aggregates the differences over the neighbors. Under a robust potential, it therefore only requires a patch to be close to the weighted mean of its neighbors rather than to each neighbor (Figure 1, lower right).

Robust potential. Connected patches truly correspond only at some positions, and residual coefficients from misaligned or different content are far larger than those from corresponding content. The gradient of a quadratic potential is proportional to the residual, so during training these large residuals dominate the gradient of the regularizer. For the image, they force dissimilar content together and oversmooth the reconstruction. For the graph, they likewise dominate the update of $\boldsymbol{\phi}$. We therefore apply the Huber potential (Huber, 1964) to the residual coefficients (Figure 1, lower right):

$$\rho_\delta(z) = \begin{cases} \frac{1}{2} z^2, & |z| \leq \delta, \\ \delta \left(|z| - \frac{1}{2}\delta \right), & |z| > \delta. \end{cases} \tag{6}$$

Its derivative $\psi_\delta(z) = \max(-\delta, \min(\delta, z))$ coincides with that of the quadratic potential for $|z| \leq \delta$, so reliable correspondences receive the full constraint. For $|z| > \delta$, the derivative is clipped at δ, so the contribution of any single coefficient to the gradient is bounded. Wrong edges thus cannot dominate the update of the image or of the graph.

Transform domain. The robust potential clips coefficient by coefficient, so the basis in which clipping takes place determines which residual components are treated as outliers. Under a quadratic

potential, an orthogonal transform leaves the energy unchanged (Parseval's identity), so this step matters only once the potential is robust. As in Figure 1 (lower right), we decompose each residual patch with a one-level 2D orthogonal Haar transform $\mathcal{H}$ (Mallat, 1989) over 2×2 cells into a low-frequency subband LL and detail subbands LH, HL, HH (p even). We then apply the Huber potential to the coefficient $\mathcal{H}_b(\cdot)_\kappa$ of subband b at cell κ. For example, let similar patches differ by a uniform brightness offset Δ. In the pixel domain, the offset is added to all p^2 residual coefficients. This pushes the coefficients that carry structural differences past the threshold as well, so their gradients are clipped. In the Haar domain, the offset enters only the $p^2/4$ coefficients of LL and is clipped and absorbed there, leaving the structural constraints on the detail subbands unaffected.

Combining the three steps gives the regularization energy of LNRR (Figure 1, bottom), in which $\boldsymbol{\phi}$ determines which patches constrain each other and $\boldsymbol{\theta}$ determines the content being constrained:

$$\mathcal{R}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \frac{1}{C} \sum_{c=1}^{C} \operatorname*{mean}_{k,b,\kappa} \rho_\delta \left(\mathcal{H}_b \left(\mathbf{E}^{(c)}_{\boldsymbol{\theta},\boldsymbol{\phi}} \right)_{k,\kappa} \right), \qquad b \in \{LL, LH, HL, HH\}, \tag{7}$$

where the mean runs over the cells inside the image (Appendix A.1).

3.4 Training and Inference

Substituting Eq. (7) into Eq. (1), we jointly solve $\min_{\boldsymbol{\theta},\boldsymbol{\phi}} \mathcal{L}_{\text{obs}}(\boldsymbol{\theta}) + \lambda \mathcal{R}(\boldsymbol{\theta}, \boldsymbol{\phi})$ for the current image, with one Adam optimizer per network. The gradient of $\boldsymbol{\phi}$ comes only from the regularizer (the dashed arrow $\nabla_{\boldsymbol{\phi}}$ in Figure 1). The energy is evaluated in low-rank form without forming the $N \times N$ adjacency, so its memory grows linearly with N (Appendix A.2). At inference, $g_{\boldsymbol{\phi}}$ is discarded, and the backbone INR is unchanged.

4 Experiments

4.1 Experimental Setup

Data and masks. We use 36 color images: Classic6 (six images from Set5 and Set14, Bevilacqua et al., 2012; Zeyde et al., 2012), Kodak10, Urban10, and DIV2K10, which take 10 images each from Kodak24, Urban100 (Huang et al., 2015), and DIV2K (Agustsson & Timofte, 2017). The longer side is resized to at most 256 pixels. The five missing patterns are *patch* (random squares, 20%), *texture* (irregular blobs, 30%), *random* (pixel-wise, 75%), *column* (50%), and *mixture* (the other four in the four quadrants, 44%) (Appendix B.1).

Compared methods and implementation. All regularizers act on the same image INR, a SIREN with 4 layers of width 256. We compare LNRR with no regularization, TV, INRR (Li et al., 2023), and the nonlocal low-rank method CRNL (Luo et al., 2025). LNRR uses $p = 16$, $s = 4$, $r = 700$, $\delta = 0.1$, and $\lambda = 0.01$, and all INR-based methods are trained with Adam (Kingma & Ba, 2015) for 20k steps (Appendix B.2). We report PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018) on the missing region at the end of training.

4.2 Image Completion

Main results. Table 1 summarizes the results with the SIREN backbone. LNRR is best on all three metrics on every mask, and in PSNR on every dataset (Appendix C.1), clearly exceeding the strongest

Table 1: **Image completion** with the SIREN backbone. Metrics are measured on the missing regions and averaged over all images and masks. Time is for one 256×256 image on an RTX 4090.

Method	PSNR (dB) ↑	SSIM ↑	LPIPS ↓	Time (s)	Params
SIREN	16.56	0.503	0.308	53	198.9K
+ TV	21.09	0.693	0.174	166	198.9K
+ INRR	21.22	0.695	0.138	215	280.1K
+ LNRR	**22.21**	**0.733**	**0.099**	199	300.2K
CRNL	21.08	0.672	0.131	13	2.02M

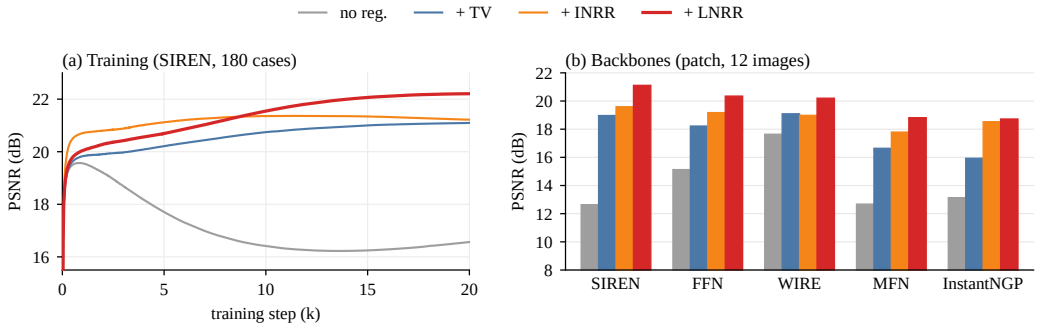

Figure 2: **Missing-region PSNR during training and across backbones.** (a) Step-wise average over all images and the five masks with the SIREN backbone. (b) Five INR backbones on the patch mask, averaged over 12 images (values in Appendix C.2).



Figure 3: **Reconstructions under three missing patterns.** Zoom-ins of the red and blue boxes are shown below each image, light green marks missing pixels, and numbers are missing-region PSNR (dB). TV, INRR, and CRNL blur or break the wing veins, the hat brim, the arched windows, and the spire, which LNRR restores.

baseline, INRR, in mean PSNR, and its training time is comparable to that of INRR. The gain over the strongest baseline of each mask depends on the geometry of the missing region. It is largest on the patch mask, where the interior of each missing square lies far from the observations, and smallest on random and column, where most missing pixels are adjacent to observations and local interpolation already recovers most of the content (Figure B1(f), Appendix C.5).

Figure 2(a) tracks the missing-region PSNR during training. Unregularized SIREN peaks early and then declines. INRR rises fastest early on, and LNRR overtakes it midway through training (Appendix C.6).

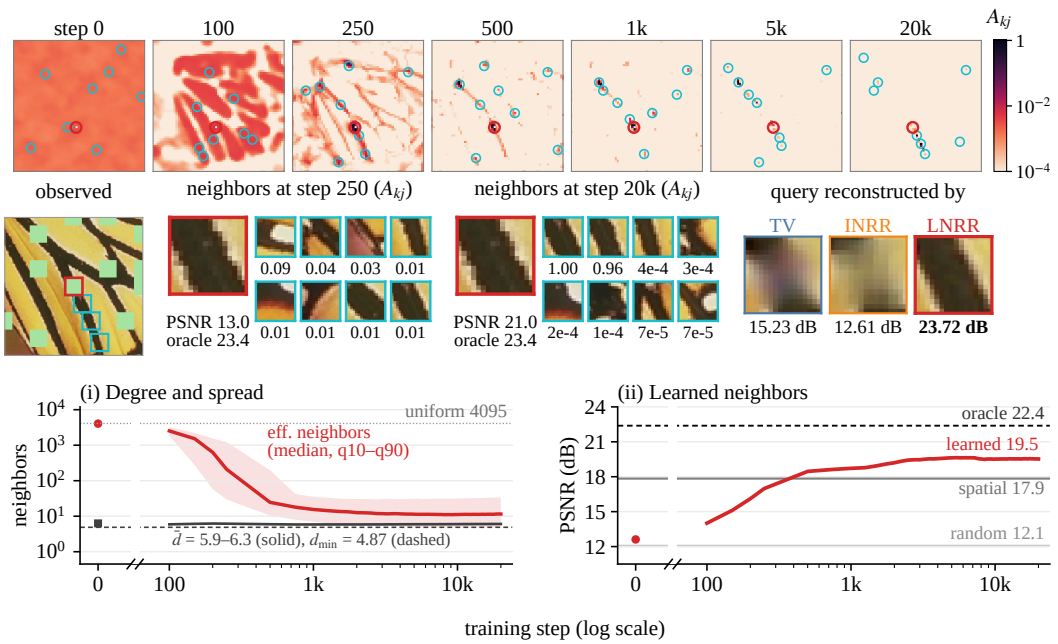

Figure 4: **Formation of the patch graph and neighbors of missing patches.** Top: adjacency row $A_{k\cdot}$ of a fully missing query patch (red, log scale) and its 8 strongest non-overlapping neighbors (cyan). Middle: the query with these neighbors at 250 and 20k steps, and the query reconstructed by TV, INRR, and LNRR. (i) Mean degree and effective number of neighbors. (ii) PSNR of the neighbor composite for query patches with $\geq 50\%$ missing.

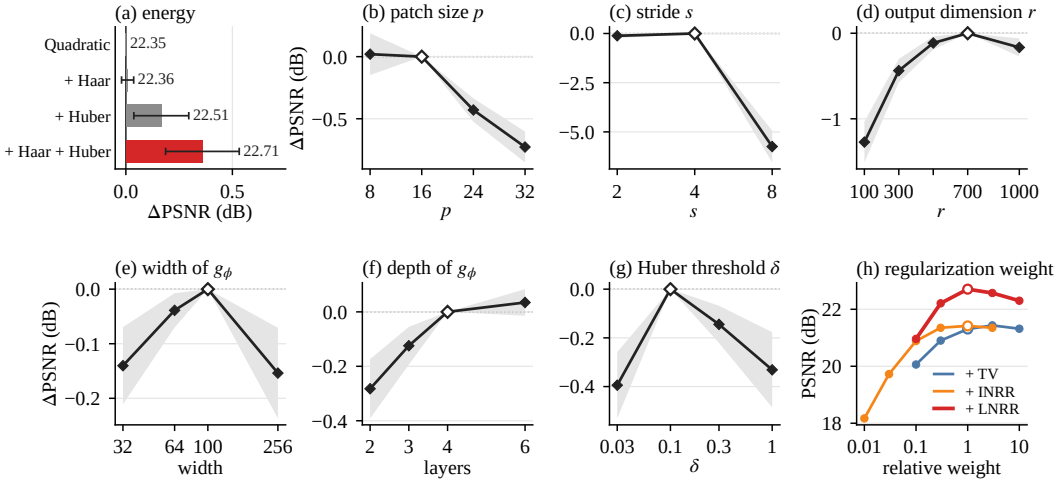

Figure 5: **Ablation and parameter analysis** (36 cases). (a) PSNR gain over the pixel-domain quadratic potential, with the mean PSNR at bar ends. (b)–(g) PSNR change from the default setting (open markers). (h) PSNR against the regularization weight relative to that of Table 1. Error bars and shading: 95% confidence intervals.

Other backbones. On four other INR backbones under the patch mask (Figure 2(b), 12 images), LNRR also gives the highest mean PSNR.

4.3 Mechanism Analysis

Formation of the patch graph. We follow one training run of LNRR (Figure 4). At initialization, the adjacency is nearly uniform. Early in training, it concentrates on a small number of effective neighbors $\exp(H(A_{k\cdot}/d_k))$ per patch, while the mean degree stays nearly constant (Figure 4(i)). The adjacency row of the query first spreads over several wing veins and then contracts to the vein containing the query.

Neighbors of missing patches. A missing patch has no observed pixels to compare, yet g_{ϕ} assigns it neighbors. We measure their quality by the PSNR between the ground-truth query patch and the composite of the other patches weighted by A_{kj}. For query patches with $\geq 50\%$ missing, this PSNR starts at the level of random patches, soon exceeds that of the spatially nearest patches, and remains below that of the oracle patches chosen with the ground truth (Figure 4(ii)). For the query on a black vein, the strongest neighbors become veins of the same orientation, and LNRR restores the vein.

4.4 Ablation Study

Figure 5(a) starts from the graph residual with the pixel-domain quadratic potential $\rho(e) = \frac{1}{2}e^2$ and adds the Haar transform and the Huber potential one at a time. All variants use the same λ. The ablation and the parameter analysis use 12 images under the patch, texture, and mixture masks (36 cases; Appendix B.1).

With the quadratic potential, the Haar transform leaves the result almost unchanged (Parseval's identity). The Huber potential improves PSNR, and adding the Haar transform on top of it improves PSNR further. The Haar transform thus helps only together with the Huber potential. The full energy gives the highest PSNR, and its gain over the pixel-domain quadratic potential comes mainly from texture and mixture (Appendix C.4).

4.5 Parameter Analysis

Patch size and stride. Patches smaller than the default perform similarly, and larger patches lower PSNR (Figure 5(b)). A stride of 2 produces about 4 times as many patches and performs similarly. At $s = 8$, the number of patches in most images approaches r, the patch graph becomes nearly edgeless, and PSNR drops sharply (Figure 5(c)).

Patch feature network. Since $\mathbf{G}_{\phi}\mathbf{G}_{\phi}^{\top}$ has rank at most r, PSNR increases with r up to 700 and decreases slightly at 1000 (Figure 5(d)). The width and depth of g_{ϕ} change PSNR only slightly (Figure 5(e)(f)).

Energy. PSNR is highest at $\delta = 0.1$ and $\lambda = 0.01$ and drops by at most 0.5 dB for δ from 0.03 to 1 and λ from 0.003 to 0.1 (Figure 5(g)(h)). With their weights tuned on the same cases, TV and INRR remain about 1.3 dB below LNRR (Appendix C.6).

5 Conclusion

INRs generalize poorly when completing images with large missing regions. To exploit the nonlocal self-similarity within a single image, we propose LNRR, which learns relations between overlapping patches with a patch feature network trained jointly with the image INR and turns these relations into a robust wavelet-domain constraint on the reconstruction. Experiments on multiple datasets and missing patterns show that LNRR recovers edges and textures that local priors smooth away. LNRR changes only the regularization term and can be plugged into different INR backbones, and on all five backbones we tested it gives the best mean PSNR under the patch mask. The mechanism analysis shows that missing patches acquire similar nonlocal neighbors during training.

Limitations and future work. LNRR learns nonlocal similarity empirically, and a theoretical account of why it improves generalization to unobserved coordinates remains open. With backbones such as FINER that converge within a few thousand steps, the patch graph forms too late to help, and scheduling the two networks, for example by warming up g_{ϕ} first, may address this. Since $\mathcal{R}$ acts only on the rendered grid, extending the patch graph from 2D images to video, 3D volumes, and scientific data is a natural next step.

AI USE STATEMENT

In this work, we used generative AI tools for implementing the method and the experiment infrastructure, for designing and iterating on experimental protocols, for aggregating and reformatting result data, for interpreting experimental results, and for translating the manuscript into English. We also used generative AI tools for producing figures, drafting and polishing sections of the paper, and searching for and verifying related literature. We did not use generative AI tools to generate synthetic data or to formulate the mathematical claims, and the other categories that require disclosure do not apply to this work. We have reviewed all AI-assisted work. All reported numbers were recomputed by scripts from the saved experiment outputs, and all AI-drafted or translated text was checked by the authors against the experimental results. We take responsibility for the final content of this work, including any text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Section 4.1 specifies the datasets, the missing patterns, the networks, and the hyperparameters. Appendix A gives the patch placement, boundary handling, and training procedure, Appendix B the data, masks, and settings of all methods, and Appendices C and D further quantitative and qualitative results. All datasets are publicly available.

REFERENCES

Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1122–1131, 2017.

Pablo Arias, Gabriele Facciolo, Vicent Caselles, and Guillermo Sapiro. A variational framework for exemplar-based image inpainting. *International Journal of Computer Vision*, 93(3):319–347, 2011.

Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie-Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *Proceedings of the British Machine Vision Conference (BMVC)*, pp. 135.1–135.10, 2012.

Michael J. Black and Anand Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International Journal of Computer Vision*, 19(1): 57–91, 1996.

Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. A non-local algorithm for image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 60–65, 2005.

Jian-Feng Cai, Raymond H. Chan, and Zuowei Shen. A framelet-based image inpainting algorithm. *Applied and Computational Harmonic Analysis*, 24(2):131–149, 2008.

Hao Chen, Bo He, Hanyu Wang, Yixuan Ren, Ser-Nam Lim, and Abhinav Shrivastava. NeRV: Neural representations for videos. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 21557–21568, 2021.

Zhengyun Cheng, Ruizhe Zhang, Guanwen Zhang, Yi Xu, Xiangyang Ji, and Wei Zhou. Low-rank tensor recovery via variational Schatten-p quasi-norm and Jacobian regularization. *Neurocomputing*, 699:134312, 2026.

Mikolaj Czerkawski, Javier Cardona, Robert Atkinson, Craig Michie, Ivan Andonovic, Carmine Clemente, and Christos Tachtatzis. Neural Knitworks: Patched neural implicit representation networks. *Pattern Recognition*, 151:110378, 2024.

Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16 (8):2080–2095, 2007.

Bin Dong, Haocheng Ju, Yiping Lu, and Zuoqiang Shi. CURE: Curvature regularization for missing data recovery. *SIAM Journal on Imaging Sciences*, 13(4):2169–2188, 2020.

Weisheng Dong, Lei Zhang, Guangming Shi, and Xiaolin Wu. Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *IEEE Transactions on Image Processing*, 20(7):1838–1857, 2011.

Weisheng Dong, Lei Zhang, Guangming Shi, and Xin Li. Nonlocally centralized sparse representation for image restoration. *IEEE Transactions on Image Processing*, 22(4):1620–1630, 2013.

Emilien Dupont, Adam Goliński, Milad Alizadeh, Yee Whye Teh, and Arnaud Doucet. COIN: COmpression with implicit neural representations. In *ICLR Workshop on Neural Compression: From Information Theory to Applications*, 2021.

Abderrahim Elmoataz, Olivier Lezoray, and Sébastien Bougleux. Nonlocal discrete regularization on weighted graphs: A framework for image and manifold processing. *IEEE Transactions on Image Processing*, 17(7):1047–1060, 2008.

Rizal Fathony, Anit Kumar Sahu, Devin Willmott, and J. Zico Kolter. Multiplicative filter networks. In *International Conference on Learning Representations (ICLR)*, 2021.

Guy Gilboa and Stanley Osher. Nonlocal operators with applications to image processing. *Multiscale Modeling & Simulation*, 7(3):1005–1028, 2008.

Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2862–2869, 2014.

Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5197–5206, 2015.

Langwen Huang and Torsten Hoefler. Compressing multidimensional weather and climate data into neural networks. In *International Conference on Learning Representations (ICLR)*, 2023.

Peter J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35 (1):73–101, 1964.

Stefan Kindermann, Stanley Osher, and Peter W. Jones. Deblurring and denoising of images by nonlocal functionals. *Multiscale Modeling & Simulation*, 4(4):1091–1115, 2005.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Zhemin Li, Tao Sun, Hongxia Wang, and Bao Wang. Adaptive and implicit regularization for matrix completion. *SIAM Journal on Imaging Sciences*, 15(4):2000–2022, 2022.

Zhemin Li, Hongxia Wang, and Deyu Meng. Regularize implicit neural representation by itself. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10280–10288, 2023.

David B. Lindell, Dave Van Veen, Jeong Joon Park, and Gordon Wetzstein. BACON: Band-limited coordinate networks for multiscale scene representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16252–16262, 2022.

Yuanyuan Liu, Yuanbiao Yang, Jing Cheng, Zhuo-Xu Cui, Qingyong Zhu, Congcong Liu, Yuliang Zhu, Jingran Xu, Hairong Zheng, Dong Liang, and Yanjie Zhu. Unsupervised dynamic MRI reconstruction via low-rank patch-based tensor functions with implicit neural representation. *IEEE Journal of Biomedical and Health Informatics*, 2026. Early access.

Zhen Liu, Hao Zhu, Qi Zhang, Jingde Fu, Weibing Deng, Zhan Ma, Yanwen Guo, and Xun Cao. FINER: Flexible spectral-bias tuning in implicit NEural representation by variable-periodic activation functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2713–2722, 2024.

Yifei Lou, Xiaoqun Zhang, Stanley Osher, and Andrea Bertozzi. Image recovery via nonlocal operators. *Journal of Scientific Computing*, 42(2):185–197, 2010.

Yuzhe Lu, Kairong Jiang, Joshua A. Levine, and Matthew Berger. Compressive neural representations of volumetric scalar fields. *Computer Graphics Forum*, 40(3):135–146, 2021.

Yisi Luo, Xile Zhao, Zhemin Li, Michael K. Ng, and Deyu Meng. Low-rank tensor function representation for multi-dimensional data recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):3351–3369, 2024.

Yisi Luo, Xile Zhao, and Deyu Meng. Revisiting nonlocal self-similarity from continuous representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(1):450–468, 2025.

Stéphane G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4455–4465, 2019.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 405–421, 2020.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 41(4):102:1–102:15, 2022.

Yuan Ni, Zhantao Chen, Shizhou Xu, Cheng Peng, Rajan Plumley, Chun Hong Yoon, Jana B. Thayer, and Joshua J. Turner. Multi-resolution enhancement for full-spectrum neural representations. *Nature Machine Intelligence*, 8(9):1492–1505, 2026.

Stanley Osher, Zuoqiang Shi, and Wei Zhu. Low dimensional manifold model for image processing. *SIAM Journal on Imaging Sciences*, 10(4):1669–1690, 2017.

Jiahao Pang and Gene Cheung. Graph Laplacian regularization for image denoising: Analysis in the continuous domain. *IEEE Transactions on Image Processing*, 26(4):1770–1785, 2017.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 165–174, 2019.

Gabriel Peyré, Sébastien Bougleux, and Laurent Cohen. Non-local regularization of inverse problems. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 57–68, 2008.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 5301–5310, 2019.

Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1–4):259–268, 1992.

Vishwanath Saragadam, Daniel LeJeune, Jasper Tan, Guha Balakrishnan, Ashok Veeraraghavan, and Richard G. Baraniuk. WIRE: Wavelet implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18507–18516, 2023.

Shayan Shekarforoush, David B. Lindell, David J. Fleet, and Marcus A. Brubaker. Residual multiplicative filter networks for multiscale reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 8550–8563, 2022.

Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 7462–7473, 2020.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 7537–7547, 2020.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9446–9454, 2018.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.

Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. *Computer Graphics Forum*, 41(2):641–676, 2022.

Chang Yu, Yisi Luo, Kai Ye, Xile Zhao, and Deyu Meng. Cross-frequency implicit neural representation with self-evolving parameters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(8):9467–9485, 2026.

Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *Proceedings of the International Conference on Curves and Surfaces*, pp. 711–730, 2012.

Jian Zhang, Debin Zhao, and Wen Gao. Group-based sparse representation for image restoration. *IEEE Transactions on Image Processing*, 23(8):3336–3351, 2014.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018.

Hao Zhu, Fengyi Liu, Qi Zhang, Zhan Ma, and Xun Cao. RHINO: Regularizing the hash-based implicit neural representation. *Science China Information Sciences*, 69(1):112101, 2026.

APPENDIX

Appendix A gives method details omitted from Section 3, Appendix B the experimental setup, Appendix C additional quantitative results, and Appendix D qualitative results.

A METHOD DETAILS

A.1 PATCH GRID AND BOUNDARY HANDLING

Patch grid. Patch centers are placed with stride s on a $\lceil m/s \rceil \times \lceil n/s \rceil$ grid. The patch at grid position (k_1, k_2) covers rows $k_1 s - p/2$ to $k_1 s + p/2 - 1$ and columns $k_2 s - p/2$ to $k_2 s + p/2 - 1$ (0-indexed) and is zero-padded outside the image. In the geometric validity mask, $V_{ko} = 1$ indicates that offset o of patch k lies inside the image, regardless of whether that pixel is observed. The center coordinate $\mathbf{u}_k$ is obtained by mapping the row and column indices on the grid linearly to $[-1, 1]$. With $p = 16$ and $s = 4$, along the rows the first 4 pixels fall into 3 patches, the last $((m-1) \bmod s) + 1$ pixels into 2, the 4 pixels immediately before them into 3, and all other pixels into 4. The same holds along the columns with n in place of m. The number of patches containing a pixel is the product of the counts along the two axes, which gives 16 in the interior, 9 at the top-left corner, and 4 at the bottom-right corner. When s is even and p is a multiple of 4, the patch origin $ks - p/2$ is even, so every 2×2 cell of a patch is aligned with the even rows and columns of the image. Only images with an odd side have cells cut by the boundary. In the main table, these are Zebra and Urban10-001, -008, and -009 (Table B1). A cut cell has $\omega_{k\kappa} = \frac{1}{2}$ ($\frac{1}{4}$ for the corner cell when both sides are odd), and all other cells have $\omega_{k\kappa} \in \{0, 1\}$.

Positions outside the image are excluded from the graph residual and from the energy. With the geometric validity mask V_{ko}, the residual of Eq. (5) at offset o of patch k is

$$E_{ko} = V_{ko} \sum_j A_{kj} V_{jo} \left(P_{ko} - P_{jo}\right), \tag{A1}$$

and the energy of Eq. (7) weights each Haar cell κ by its valid fraction $\omega_{k\kappa} = \frac{1}{4} \sum_{o \in \kappa} V_{ko}$:

$$\mathcal{R}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \frac{1}{C} \sum_{c=1}^{C} \frac{\sum_{k,b,\kappa} \omega_{k\kappa}\, \rho_\delta\big(\mathcal{H}_b(\mathbf{E}^{(c)})_{k,\kappa}\big)}{4 \sum_{k,\kappa} \omega_{k\kappa}}. \tag{A2}$$

A.2 TRAINING PROCEDURE

Since self-loop terms cancel in the patch differences, Eq. (A1) can use $\mathbf{G}_{\boldsymbol{\phi}} \mathbf{G}_{\boldsymbol{\phi}}^\top$ in place of $\mathbf{A}_{\boldsymbol{\phi}}$ and is evaluated in low-rank form, with memory $O(Nr)$ instead of $O(N^2)$.

Algorithm 1. Training and inference of LNRR. The inputs are the observations $\{\mathbf{y}_{ij}\}_{(i,j)\in\Omega}$ and the hyperparameters $p, s, r, \delta, \lambda, T$.

1. Randomly initialize $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$. Determine the patch center coordinates $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_N]^\top$ by Eq. (3) and the geometric validity mask $\mathbf{V}$, which depends only on the image size. Repeat steps 2–6 for $t = 1, \ldots, T$.
2. Render $\mathbf{X}_{\boldsymbol{\theta}}$ on the full pixel grid and compute $\mathcal{L}_{\text{obs}}$ by Eq. (1). Extract the patch matrices $\mathbf{P}_{\boldsymbol{\theta}}^{(c)} = \mathcal{T}_{p,s}(\mathbf{X}_{\boldsymbol{\theta}}^{(c)})$, zero-padding out-of-bounds positions.
3. Compute the patch features $\mathbf{G}_{\boldsymbol{\phi}} = \mathrm{softmax}(g_{\boldsymbol{\phi}}(\mathbf{U})) \in \mathbb{R}^{N \times r}$, with the softmax applied row-wise.
4. Compute the graph residual $\mathbf{E}^{(c)}$ of each channel by Eq. (A1) in low-rank form, multiplying $\mathbf{G}_{\boldsymbol{\phi}}^\top$ with the stacked validity mask and patch matrices before multiplying by $\mathbf{G}_{\boldsymbol{\phi}}$, so that the $N \times N$ adjacency is never formed.
5. Reshape each residual patch to $p \times p$, apply the one-level orthogonal Haar transform to obtain the coefficients of the four subbands, and compute $\mathcal{R}(\boldsymbol{\theta}, \boldsymbol{\phi})$ by Eqs. (6) and (A2).
6. Update $\boldsymbol{\theta}$ with $\nabla_{\boldsymbol{\theta}}(\mathcal{L}_{\text{obs}} + \lambda \mathcal{R})$ and $\boldsymbol{\phi}$ with $\lambda \nabla_{\boldsymbol{\phi}} \mathcal{R}$, using two Adam optimizers.
7. After training, discard $g_{\boldsymbol{\phi}}$ and output $\mathbf{X}_{\boldsymbol{\theta}}$ at step T.

B Experimental Setup

B.1 Data and Masks

Images. In Classic6, Butterfly is taken from Set5 (Bevilacqua et al., 2012) and the other five images from Set14 (Zeyde et al., 2012). Kodak10 consists of the first 10 images of Kodak, Urban10 of the first 10 images of Urban100 (Huang et al., 2015), and DIV2K10 of 10 512×512 images taken from DIV2K (Agustsson & Timofte, 2017). Each image is downsampled by block averaging with the smallest integer factor k for which the longer side does not exceed 256 (Table B1). When a side is not divisible by k, the leftover rows or columns at the edge are cropped before downsampling.

Table B1: **Images used in the experiments.** Sizes are given as height $\times$ width, and * marks images with an odd side.

Dataset	Factor	Images and sizes
Classic6	1–3	Baboon 240×250, Barbara 192×240, Butterfly, Lenna, and Pepper 256×256, Zebra* 130×195
Kodak10	3	kodim01–10: kodim04, 09, and 10 256×170, the others 170×256
Urban10	4	001–010: 001* 161×256, 002 and 005 192×256, 003, 004, and 007 170×256, 006 256×256, 008* 143×256, 009* 199×256, 010 204×256
DIV2K10	2	00–07, 09, and 10, all 256×256

Masks. Images of the same size share one mask, which is determined by the image size and a fixed random seed, and all methods use the same mask. Let the square side be $h = \lfloor \min(m, n)/16 \rfloor$, so that $h = 16$ when the shorter side is 256. *patch* divides the image into a grid of squares of side h and sets a random 20% of them missing. When a side is not divisible by h, the leftover rows or columns at the edge remain observed. Adjacent missing squares can merge into larger missing regions. *texture* filters standard normal white noise with a Gaussian of standard deviation 1.6 pixels and sets the 30% of pixels with the largest values missing. *random* keeps a random 25% of the pixels observed. *column* keeps a random half of the columns observed. *mixture* splits the image into four quadrants at $(\lfloor m/2 \rfloor, \lfloor n/2 \rfloor)$. The top-left quadrant has 75% pixel-wise random missingness, and the top-right quadrant has 30% missingness generated as in texture. The bottom-left quadrant sets 20% (rounded up) of the squares on a grid of side h missing, and any two missing squares are separated by at least one square, also in the diagonal direction. The bottom-right quadrant removes a random half of the columns. Figure B1 shows the five masks.

The ablation, the parameter analysis, and Appendices C.2–C.7 use a subset of 12 images (Butterfly, Lenna, and Zebra from Classic6, kodim01–03, Urban10 001–003, and DIV2K10 00–02) under the patch, texture, and mixture masks, which gives 36 cases. Appendix C.2 uses only the patch mask, and the top row of Figure C2 uses all 180 cases.

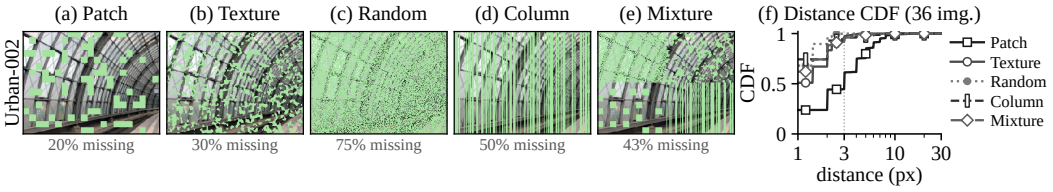

Figure B1: **The five masks.** Panels (a)–(e) show the masks on Urban10-002, with missing pixels in light green and the missing rates below; panel (f) shows the distribution of the distance from missing pixels to the nearest observed pixel.

B.2 Networks, Baselines, and Hyperparameters

Table B2 lists the backbones, SIREN (Sitzmann et al., 2020), FFN (Tancik et al., 2020), WIRE (Saragadam et al., 2023), MFN (Fathony et al., 2021), and InstantNGP (Müller et al., 2022), and the settings of the regularizers. INRR uses $\lambda = 0.1$ in Table 1 and in Figures 2 and 3, and $\lambda = 0.01$ where noted.

Table B2: **Network architectures and hyperparameters.** Unless stated otherwise, all networks are trained with Adam for 20k steps. Each step renders all pixels and minimizes the mean squared error on the observed pixels (Eq. (1)). On the four other backbones, TV, INRR, and LNRR use the same settings as with SIREN. During training, the number of parameters is 198.9K for SIREN and +TV, 280.1K for +INRR, and 300.2K for +LNRR. At inference, all methods have 198.9K parameters.

Component	Architecture and settings	LR	Params
SIREN	4 hidden layers of width 256; $\omega_0 = 30$ in the first layer and $\omega_0 = 1$ in the others	10^{-3}	198.9K
TV	Second-order differences as in the released code of Li et al. (2023), $\frac{1}{Cmn}\sum_c(\|\mathbf{D}_h\mathbf{X}^{(c)}\|_F + \|\mathbf{D}_v\mathbf{X}^{(c)}\|_F)$ over interior pixels (norms not squared), weight 1.0	–	–
INRR	Row and column graph networks, each a SIREN with 4 hidden layers of width 100 and 100 outputs, followed by a softmax; $\alpha = 0.85$, $\delta = 0.1$, $\lambda = 0.1$ (Figure 5(h))	10^{-4}	81.2K
LNRR	$p = 16$, $s = 4$, $r = 700$; g_ϕ is a SIREN with 4 hidden layers of width 100 (ω_0 as above), followed by a softmax; one-level Haar with equally weighted subbands; $\delta = 0.1$, $\lambda = 0.01$	2×10^{-3}	101.3K (output layer 70.7K)
FFN	256 Gaussian Fourier features $[\sin(2\pi\mathbf{Bv}), \cos(2\pi\mathbf{Bv})]$ of $\mathbf{v} \in [0,1]^2$, with $\mathbf{B}$ fixed and drawn from $\mathcal{N}(0, 10^2)$; 3 ReLU hidden layers of width 256; sigmoid output (Tancik et al., 2020)	10^{-3}	263.7K
WIRE	4 complex Gabor wavelet layers of width 181 ($256/\sqrt{2}$ complex units), $\omega_0 = 10$, scale parameter 10	10^{-3}	99.9K
MFN	4 sinusoidal filters and 3 multiplicative layers of width 256, input scale 256	10^{-2}	201.2K
InstantNGP	Hash grid with 8 levels, 2 features per level, table size 2^{15}, coarsest resolution 16, and growth factor 1.486 (the grid features belong to $\boldsymbol{\theta}$); 2 ReLU hidden layers of width 64; Adam with $\beta = (0.9, 0.99)$ and $\epsilon = 10^{-15}$	10^{-2}	180.8K

CRNL. CRNL (Luo et al., 2025) uses the default color-image parameters of the completion example released by its authors, with a patch width of 6 and 31 patches per group. It first runs INR initialization for 2000 steps and then performs patch matching and 401 steps of main optimization. Its 2.02M parameters in Table 1 sum both stages, including the core tensor, for a 256×256 image.

Metrics. All metrics are computed on the final reconstruction clipped to $[0, 1]$. Missing-region PSNR uses the mean squared error over all channels of the missing pixels. SSIM (Gaussian window, $\sigma = 1.5$) and LPIPS (AlexNet) maps are averaged over the missing pixels. Times in Table 1 are wall-clock times of one run on an otherwise idle RTX 4090 (CRNL: median of three runs), including start-up and periodic evaluation. Confidence intervals are 95% t-intervals of the per-case paired differences.

C Additional Quantitative Results

C.1 Per-Mask and Per-Dataset Results

Table C1 breaks down Table 1 by mask and dataset, and Table C2 gives SSIM and LPIPS per mask. LNRR is best in every column.

Table C1: **Missing-region PSNR (dB) per mask and per dataset.** Each mask column averages all images, and each dataset column averages the five masks.

	Mask					Dataset			
Method	patch	texture	random	column	mixture	Classic6	Kodak10	Urban10	DIV2K10
SIREN	12.04	16.47	19.81	17.50	16.99	17.27	18.35	15.50	15.42
+ TV	18.21	21.49	22.68	21.58	21.52	21.41	23.39	19.33	20.38
+ INRR	19.21	21.70	22.20	21.45	21.54	21.19	23.80	18.98	20.90
+ LNRR	**20.33**	**22.74**	**23.30**	**22.14**	**22.51**	**22.26**	**24.45**	**20.78**	**21.36**
CRNL	18.11	21.70	22.61	21.46	21.53	21.09	23.54	19.26	20.44

Table C2: **Missing-region SSIM and LPIPS per mask** (36 images per mask). The best value in each column is in bold.

Metric	Method	patch	texture	random	column	mixture	Average
Missing-region SSIM ↑	SIREN	0.202	0.524	0.629	0.603	0.555	0.503
	+ TV	0.503	0.730	0.746	0.761	0.726	0.693
	+ INRR	0.534	0.736	0.727	0.757	0.723	0.695
	+ LNRR	**0.596**	**0.772**	**0.766**	**0.773**	**0.758**	**0.733**
	CRNL	0.470	0.713	0.733	0.738	0.706	0.672
Missing-region LPIPS ↓	SIREN	0.418	0.250	0.309	0.272	0.293	0.308
	+ TV	0.175	0.098	0.267	0.153	0.175	0.174
	+ INRR	0.121	0.078	0.222	0.125	0.141	0.138
	+ LNRR	**0.104**	**0.049**	**0.149**	**0.095**	**0.098**	**0.099**
	CRNL	0.144	0.086	0.179	0.117	0.130	0.131

C.2 Other Backbones

Table C3 lists the values of Figure 2(b). On every backbone, LNRR has the highest mean PSNR under the patch mask. INRR uses $\lambda = 0.1$ as in Table 1, and none of these runs diverged.

Table C3: **Missing-region PSNR (dB) on five INR backbones under the patch mask** (12 subset images). The best value in each row is in bold.

Backbone	None	+ TV	+ INRR	+ LNRR
SIREN	12.64	18.97	19.60	**21.11**
FFN	15.13	18.22	19.18	**20.35**
WIRE	17.64	19.09	18.98	**20.20**
MFN	12.68	16.65	17.79	**18.82**
InstantNGP	13.14	15.94	18.53	**18.73**

C.3 Comparison with CRNL on the Subset

Table C4 compares CRNL (Luo et al., 2025), which likewise uses only the observations of a single image, with the three regularizers on the SIREN backbone over the 36 subset cases. The missing-region PSNR of LNRR is 1.45 dB higher than that of CRNL (95% confidence interval $[1.12, 1.79]$) and is higher on all three masks, and missing-region SSIM and LPIPS also favor LNRR.

Table C4: **Comparison with CRNL on the subset.** Results cover the 36 subset cases. PSNR is the missing-region PSNR (dB), and SSIM and LPIPS are averaged over the missing regions of the 36 cases. INRR uses $\lambda = 0.1$ as in Table 1. The best value in each column is in bold.

Method	patch	texture	mixture	Average	SSIM ↑	LPIPS ↓
SIREN + TV	18.97	22.54	22.41	21.31	0.666	0.146
SIREN + INRR	19.60	22.50	22.15	21.42	0.671	0.113
SIREN + LNRR	**21.11**	**23.65**	**23.35**	**22.71**	**0.718**	**0.081**
CRNL	18.88	22.51	22.37	21.25	0.643	0.112

C.4 Ablation Details

Table C5 gives the energy ablation of Figure 5(a) per mask. Table C6 keeps the energy and changes only the source of the adjacency. The learned graph is the best in every column. Non-learned nearest-neighbor graphs of the same mean degree are 1.0–1.7 dB lower, a graph frozen at initialization is 5.7 dB lower, and removing the non-overlapping (nonlocal) edges costs 0.35 dB.

Table C5: **Energy ablation per mask** (missing-region PSNR, dB; 36 subset cases).

Energy	Haar	Huber	patch	texture	mixture	Average
Pixel-domain quadratic potential			21.02	23.17	22.85	22.35
+ Haar	✓		21.08	23.14	22.84	22.36
+ Huber		✓	21.00	23.37	23.17	22.51
+ Haar + Huber (LNRR)	✓	✓	**21.11**	**23.65**	**23.35**	**22.71**

Table C6: **Effect of the adjacency source on missing-region PSNR (dB).** The first four columns use the 36 subset cases, and the last column uses 1/8 squares (12 cases). The neighbor graphs connect each patch to its 8 most similar patches, measured on the observed pixels or on the current reconstruction, with the same mean degree as the learned graph, and the current-reconstruction graph is rebuilt every 1000 steps. In the rows with overlapping or non-overlapping edges only, g_ϕ is still trained on the full graph, and the image is constrained only by the corresponding part of the learned adjacency, rescaled to the mean degree of the full graph. The uniform graph uses $r = 1$.

Adjacency	patch	texture	mixture	Average	patch (1/8 squares)
Learned graph (LNRR)	**21.11**	**23.65**	**23.35**	**22.71**	**18.35**
Frozen after initialization	15.21	17.72	18.15	17.02	14.87
Observed-pixel neighbor graph	19.03	22.32	21.77	21.04	16.99
Current-reconstruction neighbor graph	19.61	22.76	22.70	21.69	16.77
Overlapping edges only	20.75	23.33	23.00	22.36	–
Non-overlapping edges only	19.38	22.69	22.65	21.58	–
Uniform graph ($r = 1$)	14.78	14.92	14.73	14.81	–

C.5 Size of Missing Regions

Figure C1 varies the side of the missing squares under the patch mask at a fixed missing rate. LNRR remains the best at every size, with gains over the best baseline of 1.42, 2.08, 1.19, and 0.31 dB at 1/32, 1/16, 1/8, and 1/4 of the shorter side.

C.6 Weights, Training Length, and Random Seeds

With the weights of TV and INRR tuned on the 36 subset cases (Figure 5(h)), LNRR remains 1.27 and 1.29 dB above them. Figure C2 shows the training curves. Trained for 50k steps (INRR with $\lambda = 0.01$), LNRR remains 0.87 dB above TV and 0.98 dB above INRR. Table C7 shows that three random seeds give the same ranking.

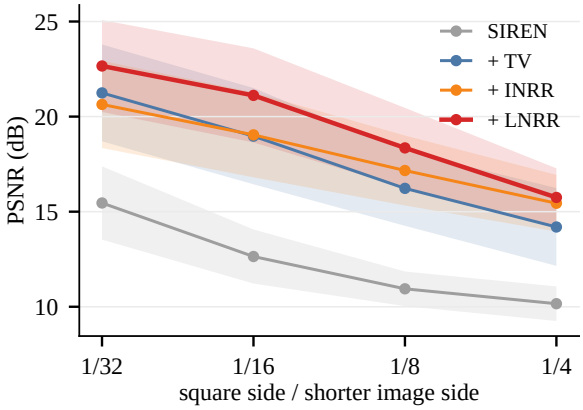

Figure C1: **Missing-region PSNR against the square size** of the patch mask (12 subset images, $\lambda = 0.01$ for INRR). Shading shows 95% confidence intervals.

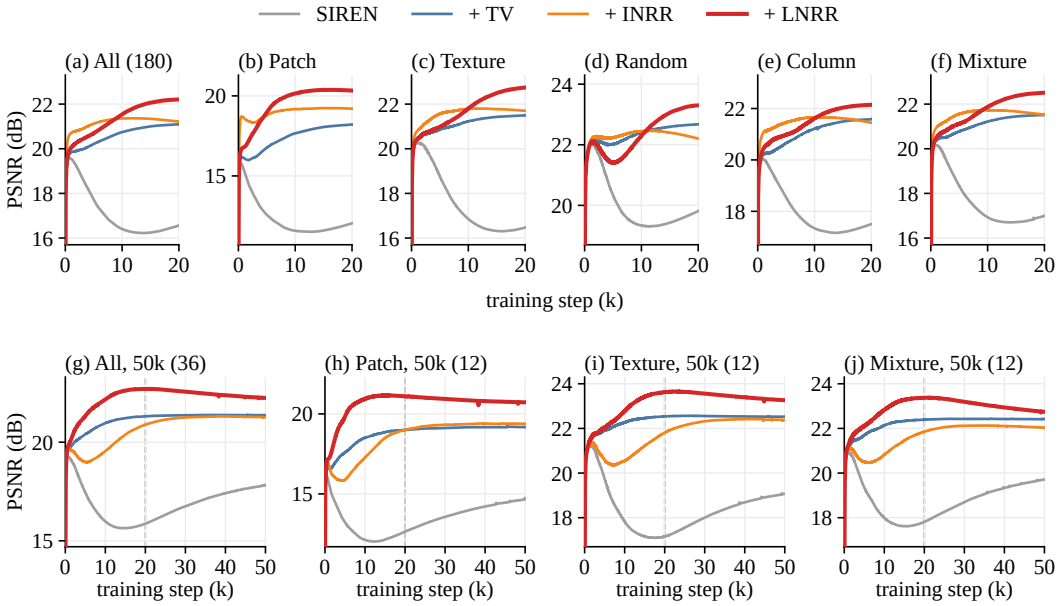

Figure C2: **Missing-region PSNR during training.** The top row uses 20k training steps and shows, from left to right, the average over the 180 main-table cases and the averages over the 36 cases of each of the five masks. The bottom row uses 50k training steps and shows, from left to right, the average over the 36 subset cases and the averages over the 12 cases of each of patch, texture, and mixture. The vertical dashed line marks step 20k. INRR uses $\lambda = 0.1$ in the top row and $\lambda = 0.01$ in the bottom row.

Table C7: **Missing-region PSNR (dB) under three random seeds.** Results cover the 36 subset cases, and the seed determines the initialization of both networks. INRR uses $\lambda = 0.01$. The per-case std is the standard deviation over the three seeds of each case, averaged over the 36 cases.

Method	Seed 0	Seed 1	Seed 2	Mean $\pm$ std over seeds	Per-case std
SIREN	15.87	15.75	15.98	15.87 $\pm$ 0.11	0.20
+ TV	21.31	21.30	21.32	21.31 $\pm$ 0.01	0.04
+ INRR	20.89	20.84	20.93	20.89 $\pm$ 0.05	0.12
+ LNRR	22.71	22.72	22.73	**22.72** $\pm$ 0.01	0.07

C.7 HIGHER RESOLUTION

Table C8 resizes the 12 subset images from their source images to a longer side of 512. LNRR remains the best, 1.50 dB above the better of TV and INRR, and $r = 700$ performs better than $r = 2800$.

Table C8: **Image completion at a longer side of 512.** 12 subset images under two masks; INRR uses $\lambda = 0.01$.

Method	patch	mixture	Average
SIREN	12.12	19.34	15.73
+ TV	15.94	22.38	19.16
+ INRR	18.94	23.69	21.31
+ LNRR ($r = 700$)	**20.53**	**25.10**	**22.82**
+ LNRR ($r = 2800$)	20.27	24.47	22.37

D QUALITATIVE RESULTS

Figures D1–D5 compare all methods on one image per dataset for each mask, Figures D6 and D7 compare the regularizers on the other backbones, and Figure D8 shows typical and failure cases.

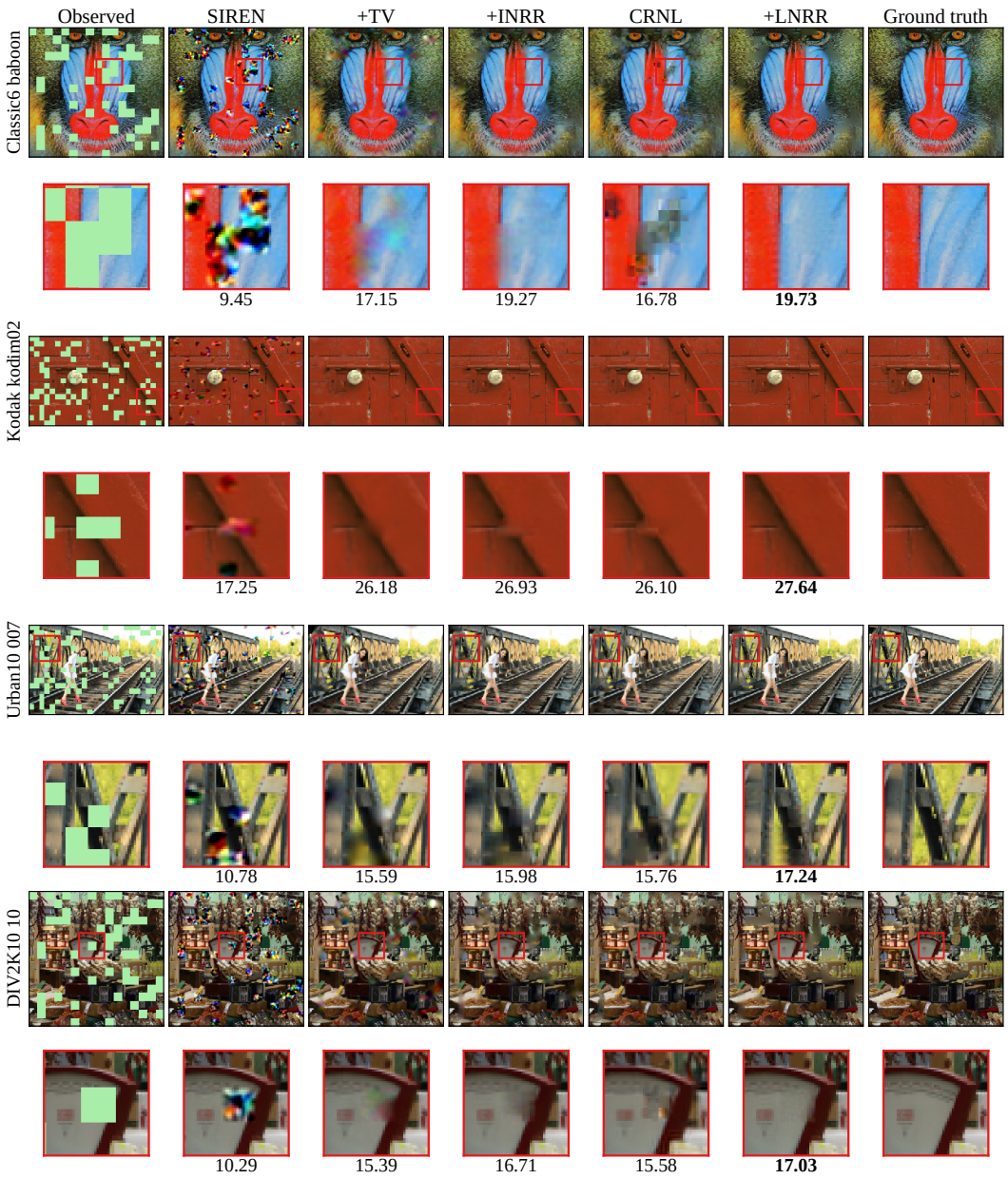

Figure D1: **Reconstructions under the patch mask.** One image per dataset, chosen as the (lower) median gain of LNRR over the best of TV, INRR, and CRNL in that dataset, excluding the cases of Figure 3. Each image is followed by the zoom-in of its red box; numbers are missing-region PSNR (dB), best in bold.



Figure D2: **Reconstructions under the texture mask.** One image per dataset, chosen as the (lower) median gain of LNRR over the best of TV, INRR, and CRNL in that dataset, excluding the cases of Figure 3. Each image is followed by the zoom-in of its red box; numbers are missing-region PSNR (dB), best in bold.

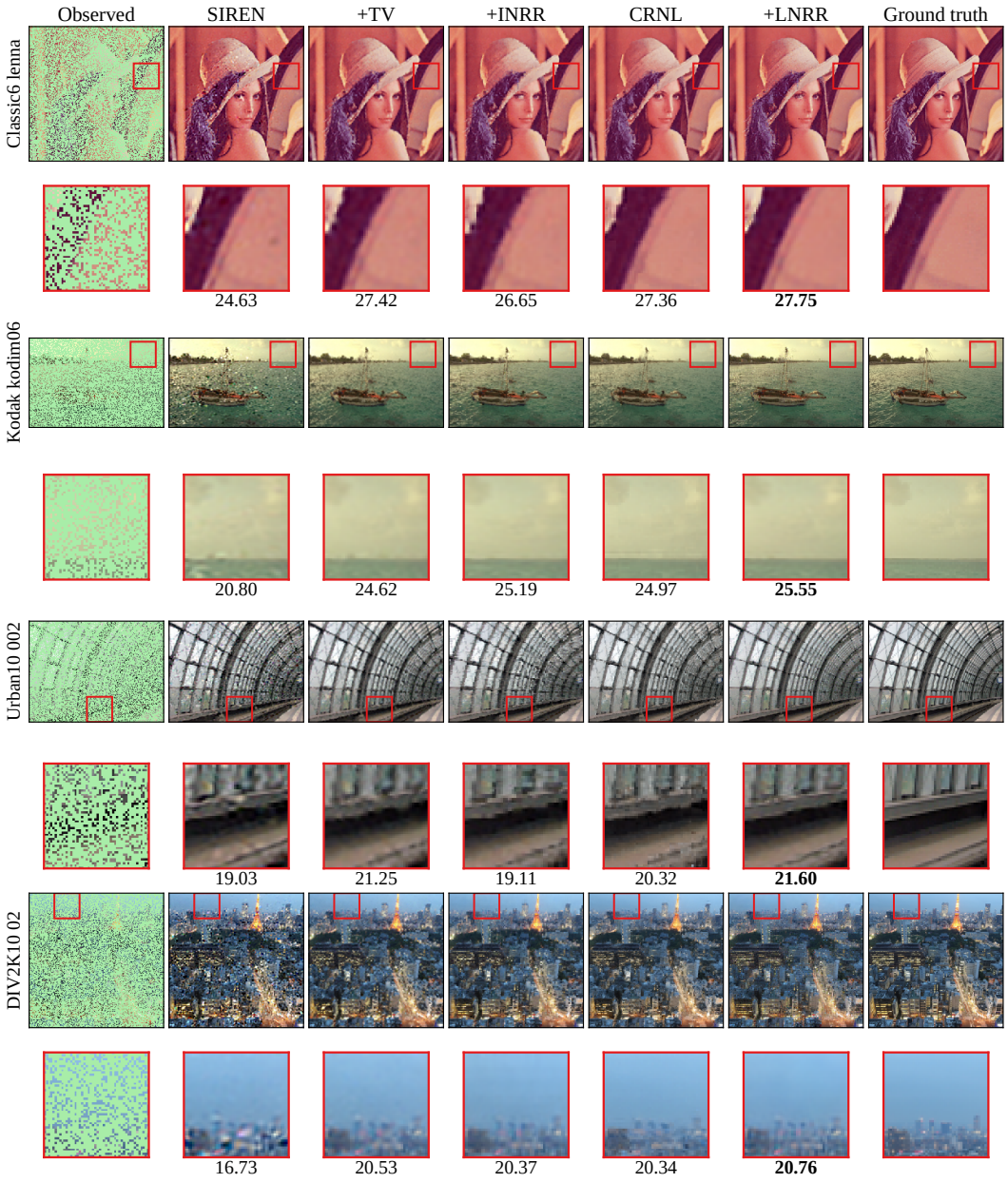

Figure D3: **Reconstructions under the random mask.** One image per dataset, chosen as the (lower) median gain of LNRR over the best of TV, INRR, and CRNL in that dataset, excluding the cases of Figure 3. Each image is followed by the zoom-in of its red box; numbers are missing-region PSNR (dB), best in bold.

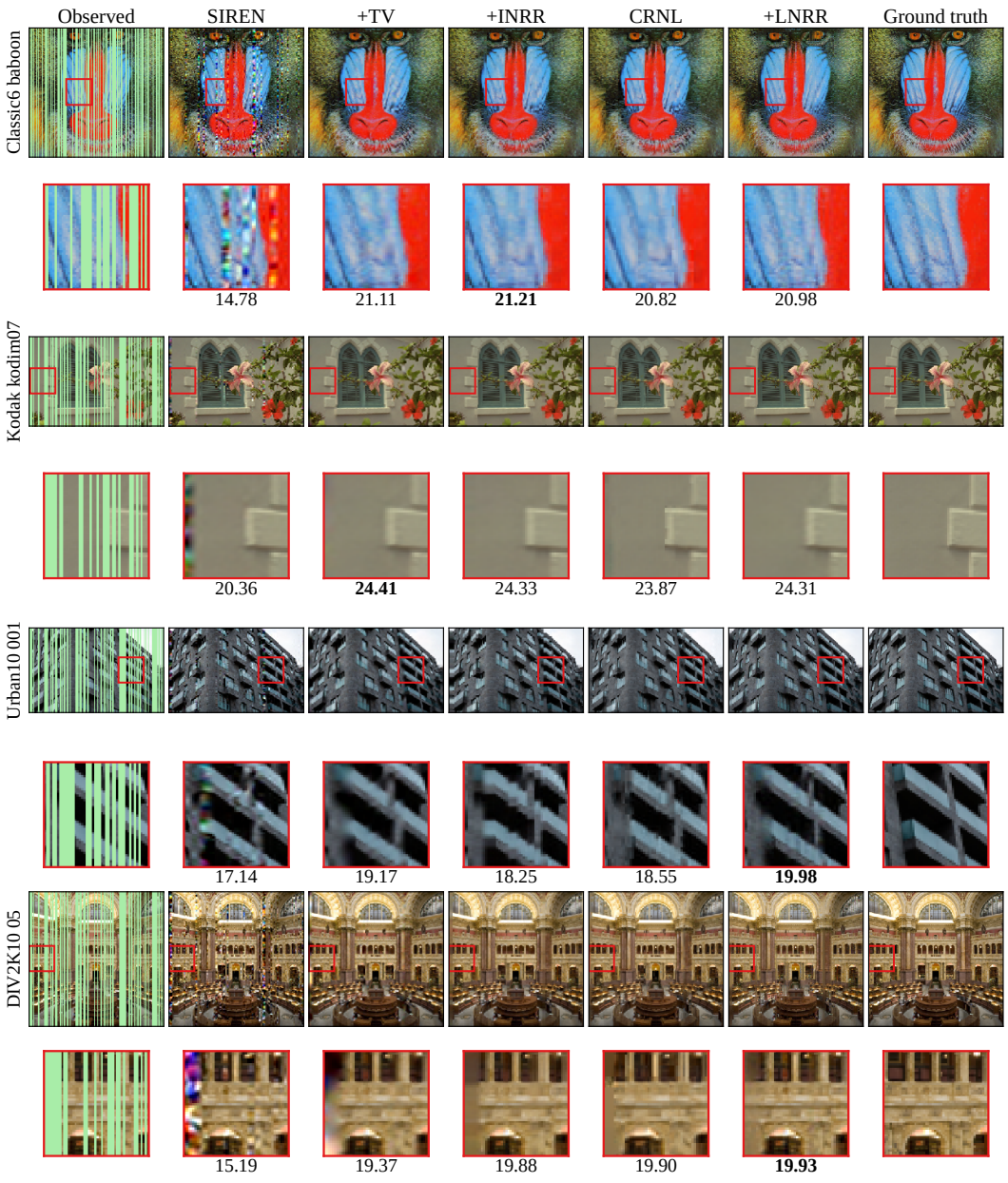

Figure D4: **Reconstructions under the column mask.** One image per dataset, chosen as the (lower) median gain of LNRR over the best of TV, INRR, and CRNL in that dataset, excluding the cases of Figure 3. Each image is followed by the zoom-in of its red box; numbers are missing-region PSNR (dB), best in bold.

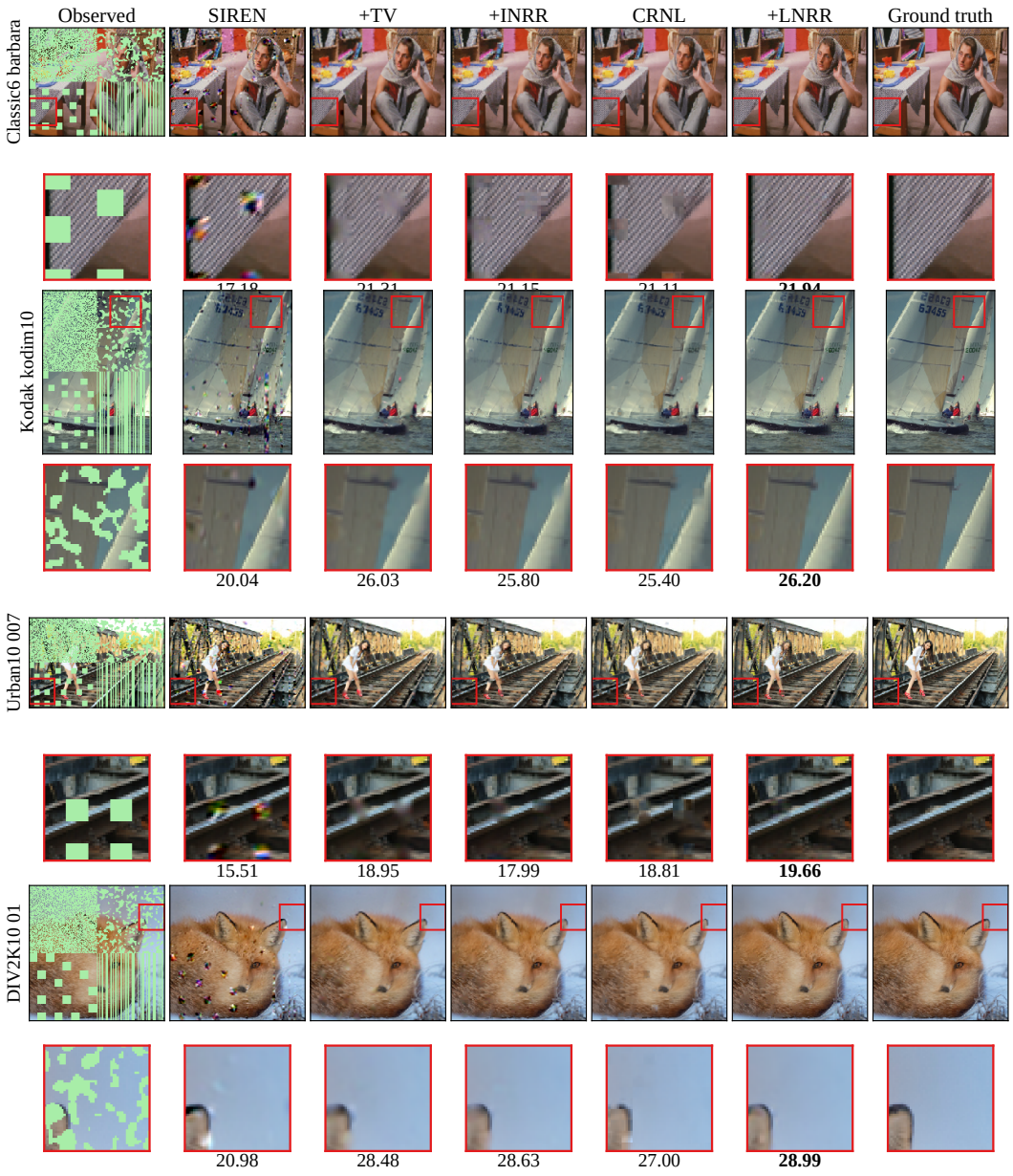

Figure D5: **Reconstructions under the mixture mask.** One image per dataset, chosen as the (lower) median gain of LNRR over the best of TV, INRR, and CRNL in that dataset, excluding the cases of Figure 3. Each image is followed by the zoom-in of its red box; numbers are missing-region PSNR (dB), best in bold.

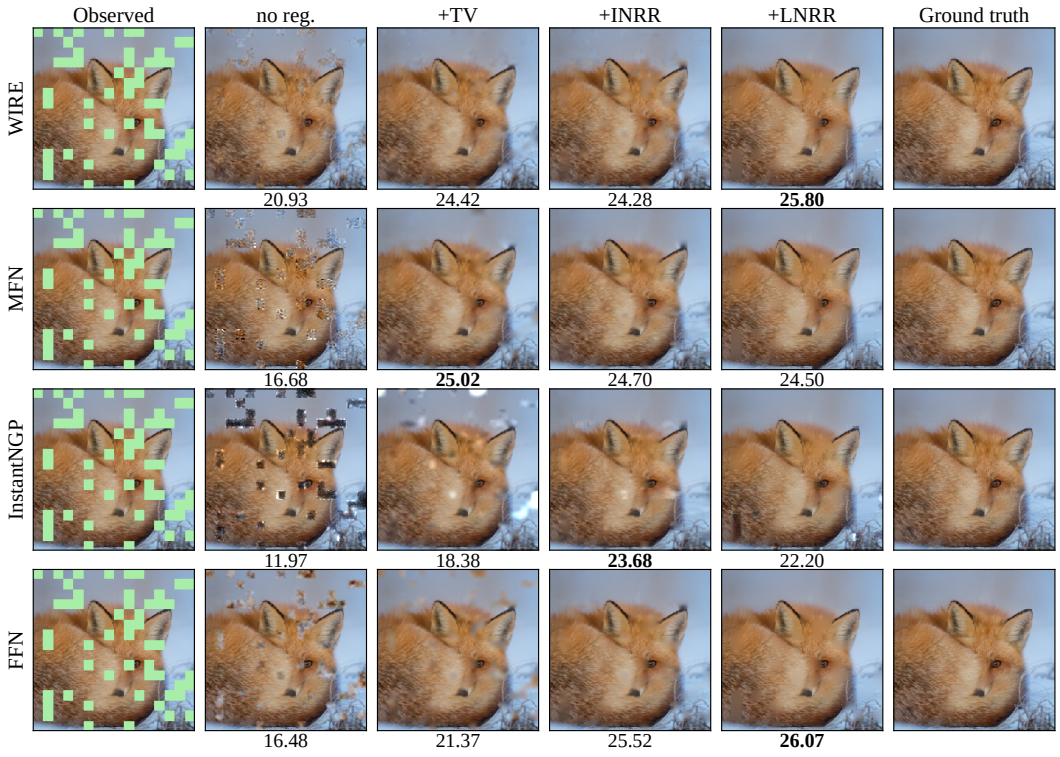

Figure D6: **Reconstructions on other backbones** (DIV2K10-01, patch). Numbers are missing-region PSNR (dB), best per row in bold.

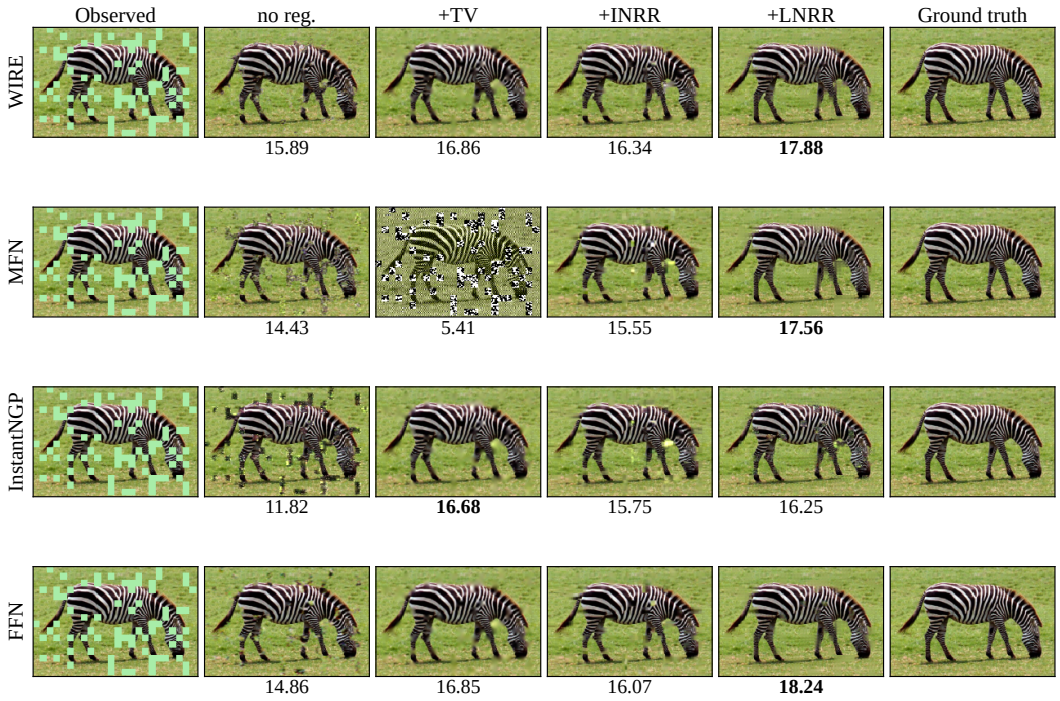

Figure D7: **Reconstructions on other backbones** (Zebra, patch). Numbers are missing-region PSNR (dB), best per row in bold.



Figure D8: **Typical and failure cases.** The first three rows show, for patch, texture, and mixture, the case whose gain of LNRR over the strongest of SIREN, TV, and INRR is the median of that mask (excluding Figure 3). The last two rows are the two cases with the smallest gain among all 180 cases, both under the column mask. Light green marks missing pixels, and numbers are missing-region PSNR (dB).