Measurement-Conditioned Raster Representation Learning: Recovering Hidden Screen Content from Diffuse Wall Reflections

Anonymous authors
Paper under double-blind review

Abstract

We ask: when a camera cannot directly capture a screen, can the screen content be recovered using only an image of the light cast by the screen onto a wall? With the display outside the camera view, diffuse reflection spreads and mixes screen light, weakening local correspondence and attenuating the directional evidence needed for small text, thin lines, and interface structure. Direct residual regression can also average foreground with background or introduce texture where the measurement is ambiguous. We therefore propose Measurement-Conditioned Raster Representation Learning (MCRRL) to learn structural correspondence between weak wall-light evidence and the hidden screen. Measurement-Conditioned Proximal Raster Fusion (MPRF) organizes directional cues through an orientation-masked dictionary and finite proximal updates, learning structural correspondence across scales. Evidence-Calibrated Digital Raster Composition Reconstruction (DRCR) uses the resulting decoder features to form latent foreground, background, and opacity fields, then uses an evidence-conditioned gate and clipping to refine their composition. Together, the two stages organize the surviving screen cues and control how they contribute to the reconstruction. Experiments on four real paired datasets cover interfaces, charts, dense glyph patterns, and browser layouts. MCRRL outperforms the compared reconstruction methods in PSNR, SSIM, and LPIPS across all four datasets. The recovered images reveal brightness distributions, regional layouts, principal contours, and portions of the displayed content. These results establish the preliminary feasibility of recovering hidden screen information from diffuse wall light with an ordinary RGB camera.

1 Introduction

A screen can be hidden from view while its light remains visible on a nearby wall. Can an ordinary RGB camera use that light to recover what the screen displays? Backes et al. (2008) recovered screen text from specular reflections in everyday objects such as eyeglasses and teapots. We extend screen-content recovery to diffuse wall reflections, reconstructing layouts and portions of foreground structure from the weak, spatially mixed light recorded by an ordinary RGB camera. Our goal is to recover the screen's visual organization and structural content from that indirect observation.

Diffuse reflection preserves the screen's light while scrambling its spatial organization. Contributions from distant screen regions overlap on the wall, and text strokes, chart lines, icon contours, and interface boundaries spread into low-contrast intensity variations. This mixing makes local wall appearance an unstable guide to local screen structure, consistent with the ill-conditioned transport studied in passive image reconstruction (Geng et al., 2022; Zhang et al., 2025). The difficulty then appears at two stages of learning. Feature extraction must preserve and combine the weak structural cues that remain across scales. Reconstruction must separate foreground from background without turning ambiguous evidence into blurred boundaries or spurious detail. These two requirements motivate the correspondence-learning and composition-reconstruction stages of MCRRL.

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.

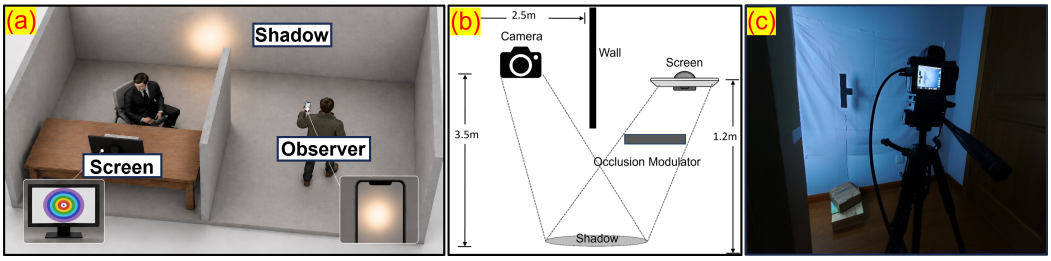

Figure 1: Scene context for hidden-screen reconstruction. (a) Illustration of the hidden-screen setting. (b) Observation geometry. (c) Physical capture setup. The screen is outside the camera's direct view; the RGB wall-light pattern supplies the observation for reconstruction.

MCRRL connects these requirements through a continuous path from wall evidence to screen structure. Measurement-Conditioned Proximal Raster Fusion (MPRF) organizes weak directional cues across encoder scales, using observation-conditioned sparse updates to build structural correspondence. Evidence-Calibrated Digital Raster Composition Reconstruction (DRCR) then reconstructs latent foreground–background composition and refines it through an evidence-guided, clipped update. Figure 1 summarizes the observation setting. Across four types of screen content, the resulting reconstructions recover the main layout, contours, and portions of the displayed information, while leading the PSNR, SSIM, and LPIPS comparison. The experiments therefore answer the opening question: an ordinary RGB observation of wall light can support the preliminary recovery of a hidden screen.

The main contributions are summarized as follows.

(1) We propose MCRRL for recovering hidden screen content from diffuse wall reflections, learning the connection between weak indirect measurements and digital screen structure.

(2) We introduce MPRF to organize observation-conditioned directional evidence across scales and DRCR to reconstruct foreground–background composition through a gated, clipped update.

(3) We demonstrate the preliminary recovery of screen layout and structural content on four real paired datasets, outperform the compared methods in the three primary metrics, and examine this recovery under controlled changes in the wall observation.

2 Related Work

Relevant work spans indirect hidden-scene sensing, which establishes what survives indirect illumination, and learned restoration components for representing and constraining ambiguous measurements.

2.1 Indirect and Passive Hidden-Scene Sensing

Indirect imaging establishes how hidden-scene information can be recovered from light reaching visible surfaces. Time-resolved, steady-state, and thermal approaches provide different measurement routes to this problem (Velten et al., 2012; O'Toole et al., 2018; Lindell et al., 2019; Chen et al., 2019; Maeda et al., 2019). Occlusion-assisted methods exploit directional information in shadows and penumbrae (Torralba & Freeman, 2012; Bouman et al., 2017); Saunders et al. (2019) recover hidden two-dimensional scenes with an ordinary camera, and Czajkowski & Murray-Bruce (2024) use two occluding edges for three-dimensional full-color reconstruction. Learning also recovers hidden motion through inverse light transport (Aittala et al., 2019) and infers people or activities from wall-illumination changes (Sharma et al., 2021). For image reconstruction, Geng et al. (2022) connect projection and hidden-image representations through manifold embedding and optimal transport, while Zhang et al. (2025) use a learned transport representation to modulate reconstruction across transport conditions. Screen side-channel studies establish complementary optical and acoustic routes to displayed information (Kuhn, 2002; Backes et al., 2008; Genkin et al., 2019). We focus this reconstruction problem on the organized content of digital displays: text strokes, chart lines, icon contours, and interface boundaries. MCRRL couples measurement-conditioned directional representation with evidence-guided foreground–background composition to recover these structures from diffuse wall light.

2.2 Learned Representations and Evidence-Guided Reconstruction

Deep restoration architectures provide encoder–decoder, attention, large-kernel, and frequency-aware representations (Zhang et al., 2017; Ronneberger et al., 2015; Liang et al., 2021; Zamir et al., 2022; Chen et al., 2022; Cui et al., 2024; Suvorov et al., 2022). Learned iterative methods turn sparse inference into finite trainable recursions (Gregor & LeCun, 2010; Zhang & Ghanem, 2018; Monga et al., 2021). Layer composition provides a representation of foreground–background mixtures (Porter & Duff, 1984), and Deep Image Matting learns opacity from an image and its trimap (Xu et al., 2017). Yan et al. (2025) separate chromatic information from intensity in HVI-CIDNet to learn photometric restoration for low-light images. We couple measurement-conditioned structural learning with foreground–background reconstruction in MCRRL, reaching an SSIM of 0.82416 versus 0.69875 for HVI-CIDNet in our Screen comparison (Table 2). MPRF organizes directional features through observation-conditioned sparse inference, and DRCR jointly learns foreground, background, opacity, and a spatially weighted composition update. Confidence enters this reconstruction as a learned spatial weight; probability calibration studies the agreement between predicted probabilities and empirical correctness (Guo et al., 2017).

3 Method

3.1 Overall Architecture

MCRRL couples structural correspondence learning with evidence-guided reconstruction. Measurement-Conditioned Proximal Raster Fusion (MPRF) organizes measurement-supported raster primitives across encoder scales, and Evidence-Calibrated Digital Raster Composition Reconstruction (DRCR) reconstructs latent foreground–background layers from decoder evidence through a gated, clipped composition update. A parallel correction head supplements the composition path, and the two residuals jointly form the screen prediction.

Figure 2 depicts the architecture overview of MCRRL for the reconstruction from the wall observation x to the predicted screen. Observation-derived condition features guide MPRF at the coarser encoder scales, where weak directional cues are organized before decoding. The decoder combines these features with skip-connected encoder states and produces the coarse residual r_0. DRCR uses the decoder feature, x, and $x + r_0$ to refine the foreground–background composition. We combine its residual r_c with the parallel correction r_f and the observation, then project the prediction to $[0, 1]$ for evaluation. The lower panels explain the two principal computations, and Table 3 examines how their removal or modification changes reconstruction.

3.2 Measurement-Conditioned Proximal Raster Fusion (MPRF)

MPRF gives weak wall evidence a structure suited to screen reconstruction. Learned sparse inference provides a trainable sequence of reconstruction and shrinkage steps (Gregor & LeCun, 2010; Zhang & Ghanem, 2018). We condition this sequence on the current wall observation and use horizontal, vertical, diagonal, and free-form dictionary supports to represent stroke- and boundary-like feature patterns. Spatial coefficient maps combine these patterns across locations, while the explanation-energy gate controls their contribution to the encoder feature.

The data flow starts from a main feature $F \in \mathbb{R}^{B\times C\times H\times W}$ and a learned condition feature $R \in \mathbb{R}^{B\times C\times H\times W}$ derived from a deterministic resize of the current RGB observation. As an observation model, the indirect measurement can be written as

$$x = \Gamma(y) + n, \tag{1}$$

where x and y are RGB tensors; x is the observation and y supplies paired training supervision. Inverse light-transport methods model how a hidden scene is mixed into an indirect observation (Aittala et al., 2019). Γ summarizes the mixing of screen light on the wall, and n summarizes measurement variation. We use this relation to guide feature construction: the encoder feature F and observation-derived condition R form a reference anchor, while their joint descriptor exposes the difference between the current representation and the observation-derived condition:

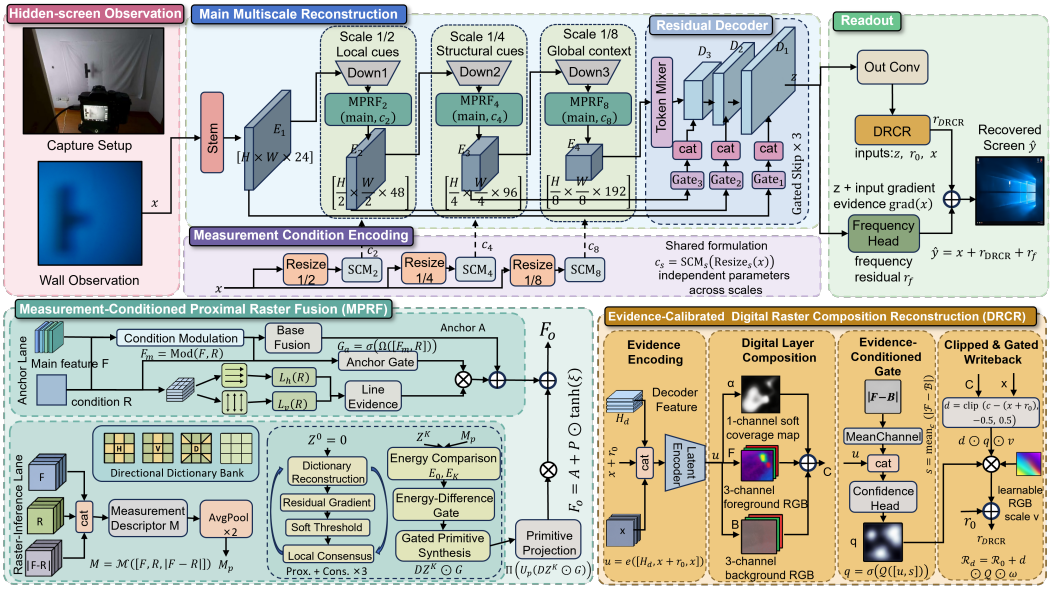

Figure 2: Overall architecture of MCRRL for hidden-screen reconstruction. The upper path maps the wall observation through a multiscale analysis–synthesis chain. Deterministically resized observations are transformed into condition states and injected at the corresponding MPRF sites, while skip transmissions return encoder evidence to the reconstruction scale. The lower-left panel expands MPRF: condition modulation builds an anchor, and a directional dictionary with proximal consensus converts measurement discrepancy into an energy-gated primitive update. The lower-right panel expands DRCR: latent foreground, background, and opacity fields form a digital-layer composition, whose evidence confidence controls a gated, clipped composition update. The final screen estimate combines the observation, the DRCR residual, and the parallel correction residual.

$$A = \mathcal{A}(F, R), \qquad M = \mathcal{M}([F, R, |F - R|]). \tag{2}$$

Here, $\mathcal{A}$ denotes the learned conditional anchor, including its condition modulation and gated detail path. The descriptor M preserves the main feature, the condition, and their absolute difference, thereby exposing measurement condition and feature–condition discrepancy to the sparse representation. The sparse surrogate is defined on reduced descriptor $P(M)$ and normalized orientation-masked dictionary D:

$$\min_{Z} \ \tfrac{1}{2}\|DZ - P(M)\|_2^2 + \lambda_k \|Z\|_1. \tag{3}$$

Here, P is a scale-reduction map, Z contains a spatial coefficient map for each dictionary atom, D is a local raster synthesis map, and step-specific $\lambda_k > 0$ controls sparsity. The dictionary is formed by masking and atom-wise normalization:

$$\bar{D}_{:,a,:,:} = D^{\mathrm{raw}}_{:,a,:,:} \odot O_{a,:,:}, \qquad D_{:,a,:,:} = \frac{\bar{D}_{:,a,:,:}}{\mathrm{clamp}_{\min}(\|\bar{D}_{:,a,:,:}\|_2, \epsilon)}, \tag{4}$$

where D^{raw} is the learned unmasked dictionary, a indexes atoms, and $O_{a,:,:}$ selects one of four local raster supports: the center row for horizontal atoms, the center column for vertical atoms, the two diagonals for diagonal atoms, and the full support for free-form atoms. The positive floor ϵ stabilizes the atom norm. The mask fixes the admissible directional support while normalization places atoms on a common explanation scale.

The sparse objective is approximated from the zero code $Z^0 = 0$ by a finite unrolled proximal-gradient recursion. Each step updates the coefficients using the gradient of the quadratic reconstruction term, then applies soft thresholding to promote a sparse representation of the directional evidence.

$$Z^0 = 0, \qquad Z^{k+1} = \mathcal{S}_{\theta_k}\left(Z^k - \eta_k D^T (DZ^k - P(M))\right), \qquad \mathcal{S}_\theta(v) = \mathrm{sign}(v)\,\mathrm{ReLU}(|v| - \theta). \tag{5}$$

The synthesis map D maps coefficients to local raster evidence and its paired adjoint D^T maps the explanation residual back to coefficient space. Here $k = 0, \ldots, K-1$ indexes the

unrolled steps. Learned η_k sets the step size, and $\theta_k = \eta_k \lambda_k$ sets the soft threshold. At each independent MPRF site, the coefficient field receives a bounded signed consensus correction formed from a pooled and upsampled coefficient copy:

$$Z^{k+1} \leftarrow Z^{k+1} + \rho_k \big(U_c(P_c(Z^{k+1})) - Z^{k+1}\big). \tag{6}$$

Here, P_c and U_c are coarse-to-fine scale maps, and bounded learned scalar ρ_k controls the signed correction. The recursion then compares spatial explanation energies:

$$E_0 = \mathrm{mean}_c(P(M)^2), \qquad E_K = \mathrm{mean}_c\left((DZ^K - P(M))^2\right), \tag{7}$$

where mean_c averages over descriptor channels, and K is the final recursion index. The sparse update is weighted by how well it explains the observation-conditioned descriptor. MPRF compares the initial descriptor energy with the residual energy after the proximal updates and converts their signed relative difference into a spatial gate. This gate adjusts the contribution of the synthesized directional features before they are returned to the encoder grid and added to the anchor.

$$G = \sigma\left(\mathrm{softplus}(\gamma)\frac{E_0 - E_K}{\mathrm{clamp}_{\min}(E_0, \epsilon)} + \beta\right), \qquad F_o = A + \Pi\big(U(DZ^K \odot G)\big) \odot \tanh(\xi). \tag{8}$$

Here, G is a one-channel spatial gate on the reduced grid, γ and β are learned gain and bias parameters, and $\epsilon > 0$ is a numerical floor. The synthesized state is returned to the descriptor grid by U, mapped by Π, and scaled through the zero-initialized injection factor $\tanh(\xi)$ before being added to anchor A.

The dictionary, recursion parameters, consensus coefficients, and readout scales are optimized end to end under paired reconstruction supervision. The gated update reaches the decoder through encoder features and skip connections. The decoder reconstructs the screen through DRCR and a parallel correction head.

The lower-left panel of Figure 2 links the discrepancy descriptor to the sparse correction and its local explanation gate. Chi et al. (2020) combine spatial convolutions with patch- and image-level Fourier processing in Fast Fourier Convolution (FFC) to aggregate local and global information. In our fixed-U-Net study, MPRF organizes the remaining wall cues through observation-conditioned directional inference and yields a smaller relative LPIPS increase than FFC at all three tested nonzero partial-support-loss levels (Figure S34).

3.3 Evidence-Calibrated Digital Raster Composition Reconstruction (DRCR)

Recovering screen structure requires foreground strokes and boundaries to remain distinct from their background. Xu et al. (2017) estimate foreground opacity from an image and a trimap in Deep Image Matting, then refine the predicted matte at object boundaries. In DRCR, we learn latent foreground and background colors together with opacity from wall-conditioned decoder features under paired screen supervision. The network first forms a continuous output state; projection to $[0, 1]$ supplies the normalized prediction used by the structural losses and evaluation.

The module receives the decoder feature $H_d \in \mathbb{R}^{B\times C\times H_0\times W_0}$, the coarse RGB residual r_0, and the original observation x. The feature H_d descends from the trunk decoder and therefore carries the recalibrated encoder evidence produced by the preceding MPRF sites. The latent encoder conditions on that feature, the coarse state, and the original observation:

$$u = e([H_d, x + r_0, x]), \qquad \alpha = \sigma(p_\alpha(u)), \quad f = \sigma(p_f(u)), \quad b = \sigma(p_b(u)). \tag{9}$$

The learned map e maps decoder feature H_d, observation x, and coarse state $x + r_0$ to latent tensor u. The prediction maps p_α, p_f, p_b produce opacity α and RGB color fields f, b in the unit interval. Their layer composition is

$$c = \alpha \odot f + (1 - \alpha) \odot b. \tag{10}$$

Here, α is shared across the RGB components, so c is a convex per-pixel composition of the two latent color fields. The layer-separation map supplied to the learned gate is

$$s = \mathrm{mean}_c(|f - b|). \tag{11}$$

The separation map s measures the local color difference between the two predicted layers. Together with u, it determines the spatial weight q of the composition update. The layer fields specify the colors being combined, opacity specifies their mixture, and the gate controls how strongly that composition refines the coarse prediction.

The learned gate maps the latent feature and separation map to a spatial weight:

$$q = \sigma(\mathcal{Q}([u, s])) \,. \tag{12}$$

Here, $q \in (0, 1)$ is a learned scalar spatial confidence and $\mathcal{Q}$ is a learned local map. The composition difference is clipped to the fixed symmetric bound $\tau = 0.5$ and scaled by the learned RGB tensor v, initialized to 0.05:

$$d = \text{clip}(c - (x + r_0), -\tau, \tau), \qquad r_c = r_0 + d \odot q \odot v. \tag{13}$$

The gate q is shared across the RGB components. Clipping sets the scale of the composition difference, and the spatial gate controls its contribution at each location. The update satisfies $|r_c - r_0| \leq \tau|v|$ elementwise. The complete output and training objective are

$$\hat{y} = x + r_c + r_f, \qquad \hat{y}_c = \text{clip}_{[0,1]}(\hat{y}), \qquad \mathcal{L} = \mathcal{L}_x + \lambda_s \mathcal{L}_s + \lambda_e \mathcal{L}_e + \varrho(t)(\lambda_v \mathcal{L}_v + \lambda_f \mathcal{L}_f). \tag{14}$$

Here, r_f is the parallel gradient-conditioned RGB correction, $\hat{y}$ is the continuous prediction, and $\hat{y}_c$ is its projection to the normalized image range. The objective couples pixel accuracy with structural, boundary, perceptual, and feature-statistics constraints, so that screen reconstruction is guided by both color agreement and spatial organization. Fixed positive $\lambda_s, \lambda_e, \lambda_v, \lambda_f$ set their weights and schedule $\varrho(t)$ introduces the support terms during training. These terms and the parallel correction head remain supporting components within the two-module design.

Paired reconstruction supervision learns opacity α and confidence q as latent fields. The resulting composition, separation field, confidence, and clipped residual form the composition readout tested by the output-topology and route ablations.

The lower-right panel of Figure 2 makes the gated, clipped composition path explicit: the composed color field is compared with the coarse state, then the learned gate and clipping scale its residual.

4 Experiments and Results

Our experiments follow the recovery of hidden screen information from three perspectives. We first establish what MCRRL reconstructs across four types of displayed content, then examine how the representation and composition stages contribute to that recovery. Finally, controlled changes in the wall observation reveal how the recovered structure depends on the available optical evidence.

4.1 Experimental Setup

We evaluate independently trained models on four real paired wall-spot/screen-target datasets: Screen, Charts, Password2, and Browser, whose complete test splits contain 374, 2,000, 1,741, and 940 pairs. Table 1 summarizes the four datasets and their separate training splits. PSNR, SSIM, and MS-SSIM are higher-better; LPIPS, MSE, RMSE, and Edge-MAE are lower-better. Table 2 reports the comparison results of PSNR, SSIM, and LPIPS, with the wider metric set and results reported in the appendix.

The four encoder levels use 24/48/96/192 channels, four residual blocks each, ECA, learned skip gates, large-kernel enhancement at the first three levels, and a deepest token bottleneck, totaling 13.735M trainable parameters. Independent MPRF instances operate at 1/2, 1/4, and 1/8 scales with three proximal steps, 2× descriptor pooling, and per-step coefficient consensus. DRCR uses a two-layer gate, a 0.5 symmetric clip bound, and an RGB scale initialized to 0.05. We normalize inputs and targets to $[0, 1]$ and train each dataset independently with Adam, batch size 8, at 256×256. Training and model selection are further described in Appendix A.3.

4.2 Comparison with Existing Methods

Table 2 compares MCRRL with encoder-decoder, residual, shifted-window, efficient-attention, and activation-free restoration designs on the four fixed paired test splits (Zhang et al., 2017; Ronneberger et al., 2015; Liang et al., 2021; Zamir et al., 2022; Chen et al., 2022; Cui et al., 2024; Suvorov et al., 2022). It reports PSNR, SSIM, and LPIPS for reconstruction fidelity, structural similarity, and perceptual distance; Figure 3 shows qualitative outputs.

Dataset	Train pairs	Test pairs	Input format	Target format
Screen	2120	374	256x256 RGB	500x375 RGB; eval. 256x256
Charts	5000	2000	256x256 RGB	256x256 RGB
Password2	5000	1741	256x256 RGB	256x256 RGB
Browser	3759	940	paired RGB; eval. 256x256	paired RGB; eval. 256x256

Table 1: Training and test splits and image resolutions of the four paired datasets. Each RGB wall observation is paired with its corresponding screen target, and we train a separate model for each dataset.

Method	Source	Screen			Charts			Password2			Browser		
		PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
NLOS reconstruction and computational imaging													
LMS-NLOS	TVC'25	21.057	0.66662	0.38366	16.69	0.69237	0.50785	13.324	0.84947	0.16463	17.681	0.62529	0.46693
NLOSOT	TIP'22	24.101	0.78974	0.27171	16.627	0.70505	0.47575	8.7634	0.7949	0.42817	17.623	0.65261	0.45216
NLOST	CVPR'23	16.834	0.56131	0.4898	16.456	0.51863	0.61828	12.524	0.77391	0.3274	16.061	0.56416	0.64842
NLOSNeuS	ICCV'23	12.589	0.48096	0.59466	13.451	0.6453	0.65335	8.7703	0.79465	0.42844	8.6278	0.34738	0.74698
Transfer and attention restoration													
HVI-CIDNet	CVPR'25	20.824	0.69875	0.39833	16.332	0.69625	0.52884	12.04	0.84584	0.24215	17.024	0.62926	0.46997
AST	CVPR'24	22.063	0.69749	0.32654	17.034	0.71851	0.46326	14.477	0.83985	0.15078	17.66	0.6595	0.4432
Uformer	CVPR'22	21.653	0.70605	0.35167	17.006	0.71389	0.48997	14.6	0.88259	0.11497	17.97	0.65941	0.44111
SwinIR	ICCVW'21	14.737	0.50043	0.55238	14.552	0.64814	0.61336	9.5485	0.80028	0.33172	13.763	0.48657	0.61524
General-purpose CNN restoration													
ConvIR	TPAMI'24	21.444	0.70668	0.37912	16.756	0.7075	0.51723	14.038	0.86454	0.1541	17.767	0.65567	0.46846
NAFNet	ECCV'22	21.673	0.7076	0.33998	16.641	0.69591	0.4867	13.128	0.84662	0.17262	17.68	0.64475	0.43318
MPRNet	CVPR'21	19.515	0.6471	0.43354	16.24	0.69118	0.54394	13.319	0.86264	0.17326	17.227	0.62671	0.50177
U-Net	MICCAI'15	20.216	0.6741	0.37924	16.245	0.6858	0.50784	12.517	0.81459	0.20387	13.17	0.47829	0.59466
DnCNN	TIP'17	15.088	0.50808	0.54211	14.802	0.64195	0.5982	10.65	0.75596	0.34943	12.889	0.48564	0.58497
SCUNet	MIR'23	20.258	0.68375	0.37754	13.115	0.65163	0.65096	8.7633	0.7949	0.42824	17.083	0.63507	0.4583
CBDNet	CVPR'19	17.508	0.60418	0.45907	13.115	0.65163	0.65096	8.7633	0.7949	0.42819	7.0727	0.41034	0.81049
RIDNet	ICCV'19	17.456	0.60553	0.48243	13.115	0.65163	0.65096	8.7633	0.7949	0.42819	16.555	0.59133	0.50062
Related image degradations													
C2PNet	CVPR'23	16.304	0.55639	0.51015	15.394	0.66619	0.58848	11.169	0.80831	0.26765	16.216	0.57982	0.51823
DRS	ECCV'24	17.507	0.582	0.4883	15.626	0.67536	0.55451	12.07	0.84545	0.19725	16.039	0.56967	0.54016
AgileIR	ICANN'25	15.071	0.39984	0.5112	14.705	0.4952	0.6467	9.1061	0.7331	0.50255	12.974	0.4197	0.65104
MCRRL	Ours	24.986	0.82416	0.23283	17.14	0.72982	0.44218	17.06	0.91832	0.071859	18.258	0.67023	0.35269

Table 2: Quantitative comparison across the four paired datasets. Higher PSNR and SSIM and lower LPIPS indicate better performance. Red and blue identify the best and second-best values within each dataset–metric column. The pale-red row marks the complete model; pale-blue rows contain at least one second-best value. Values retain their reported precision.

MCRRL leads all four datasets in PSNR, SSIM, and LPIPS, connecting the recovery of screen structure to improvements in pixel, structural, and perceptual measures. Uformer combines hierarchical window attention with multiscale modulation for image restoration (Wang et al., 2022), while AST uses adaptive sparse attention and attentive feature refinement (Zhou et al., 2024). In our Password2 comparison (Table 2), MCRRL improves PSNR by 2.460 dB over Uformer and 2.583 dB over AST, reduces LPIPS by 37.5% and 52.3%, respectively, and increases SSIM from Uformer's 0.88259 to 0.91832. The agreement across these measures supports the value of organizing wall evidence for screen reconstruction. Figure 3 shows the displayed structures associated with these gains, and Tables S1–S4 extend the comparison to multiscale structure, pixel error, and edge error.

The recovered information extends from the organization of a screen to the structures within it. On Screen, MCRRL reconstructs interface panels and separates large color regions. On Charts, the outputs preserve geometric arrangements and distinct colored elements. Password2 makes the recovery of separated strokes and glyph organization directly visible, while Browser extends the result to page sections and image blocks. These examples connect the wall-light measurements to the displayed content that produced them: the reconstruction recovers both where major regions lie and how foreground structures are arranged within them.

4.3 Ablation Study

Figure 4 isolates the structural interventions on the Screen dataset, including deletion controls and same-interface replacements.

The module interventions change the foreground structures that make a screen informative. The complete model preserves the dialog and more of the icon-group and text-line

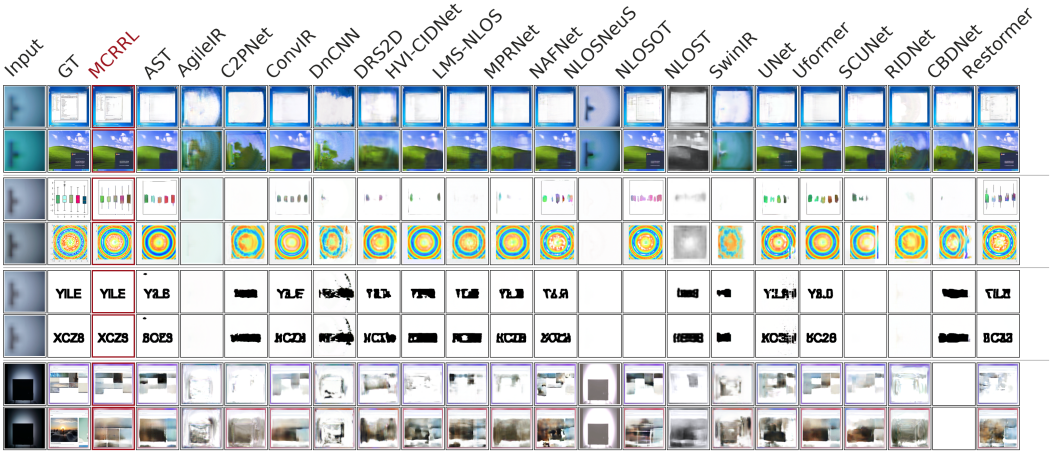

Figure 3: Main qualitative comparison across Screen, Charts, Password2, and Browser. Columns show Input, GT, MCRRL, and the 20 baseline outputs for all eight examples. MCRRL preserves brightness distributions, layout partitions, color blocks, text strokes, and principal contours under the indirect measurement.

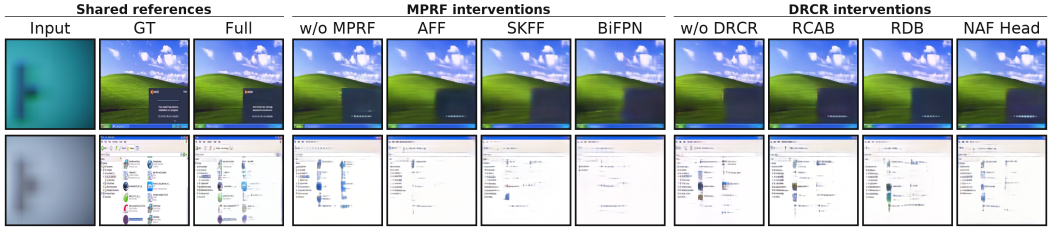

Figure 4: Screen reconstructions under MPRF and DRCR interventions. The two examples compare a desktop dialog and a dense interface layout, with shared Input, GT, and Full (MCRRL) references followed by the MPRF and DRCR intervention columns. AFF, SKFF, and BiFPN denote Attention Fusion, Selective-Kernel Fusion, and Bidirectional Pyramid; RCAB, RDB, and NAF Head denote Residual Channel Attention, Residual Dense-Block Readout, and Activation-Free Readout in Tables 3 and S5.

organization in the dense interface example. Removing or replacing the fusion and reconstruction components changes these structures, connecting the architectural comparisons to the recovery of displayed information.

Table 3 consolidates the main architectural ablations across Screen, Charts, Password2, and Browser. It combines module deletions, same-interface replacements, joint deletion, and representative internal-structure interventions; the replacement families adapt established restoration and fusion designs to the same interface (Ronneberger et al., 2015; Zhang et al., 2017; Liang et al., 2021; Cui et al., 2024). Additional internal-structure comparisons are provided in Tables S5–S8.

Screen recovery benefits from both organizing the wall evidence and controlling its conversion into foreground structure. Removing MPRF lowers Screen PSNR from 24.986 to 23.754 dB, and removing DRCR lowers it to 23.535 dB. The internal comparisons then examine how each stage contributes: coefficient consensus, repeated proximal updates, and scale placement shape the representation, while learned opacity, foreground–background composition, and spatial gating shape the reconstructed screen. On Password2, replacing these learned composition choices with fixed opacity, direct foreground prediction, or a unit gate lowers SSIM relative to the complete model. The full model leads every dataset–metric group in Table 3, supporting the combined representation-to-reconstruction design. Tables S5–S8 extend this analysis to how fusion, scale placement, conditioning, and reconstruction choices affect the recovery of screen structure.

Variant	Screen			Charts			Password2			Browser		
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Full MCRRL	24.986	0.8242	0.2328	17.14	0.7298	0.4422	17.06	0.9183	0.0719	18.353	0.6733	0.3447
w/o MPRF	23.754	0.7908	0.2772	16.789	0.7189	0.5422	15.962	0.8998	0.0952	17.258	0.6442	0.4911
Attention Fusion	23.239	0.7792	0.2972	16.219	0.7024	0.5792	15.067	0.8858	0.1205	17.156	0.6436	0.4983
w/o DRCR	23.535	0.7783	0.2847	16.823	0.7175	0.5397	15.661	0.8965	0.1022	17.119	0.6409	0.4981
Residual Channel Attention	23.398	0.7813	0.2914	16.768	0.7145	0.5434	15.451	0.8886	0.1051	17.355	0.6440	0.4944
w/o MPRF and DRCR	23.874	0.8010	0.2716	16.79	0.7128	0.5480	15.823	0.8831	0.0983	17.086	0.6368	0.5020
w/o Coefficient Consensus	23.478	0.7895	0.2914	17.016	0.7208	0.4700	16.246	0.9082	0.0847	17.956	0.6688	0.3848
Single-Step Solver	23.359	0.7841	0.2905	16.992	0.7224	0.4694	16.443	0.9111	0.0804	18.001	0.6676	0.3814
Finest-Scale-Only	23.43	0.7841	0.2936	16.974	0.7237	0.4769	16.289	0.9077	0.0836	17.943	0.6688	0.3839
Fixed Opacity	23.485	0.7898	0.2905	16.951	0.7219	0.4703	16.346	0.9088	0.0831	18.137	0.6698	0.3762
Direct Foreground	23.379	0.7888	0.2918	16.927	0.7204	0.4688	15.721	0.9040	0.0918	18.041	0.6697	0.3803
Unit Gate	23.432	0.7873	0.2916	16.921	0.7182	0.4724	16.078	0.9061	0.0836	18.019	0.6681	0.3821

Table 3: Main ablation study across the four paired datasets. Each dataset group reports PSNR, SSIM, and LPIPS. Red and blue indicate the best and second-best values within each dataset–metric column. Training and model selection are described in Appendix A.3. Additional internal-structure comparisons are provided in Tables S5–S8.

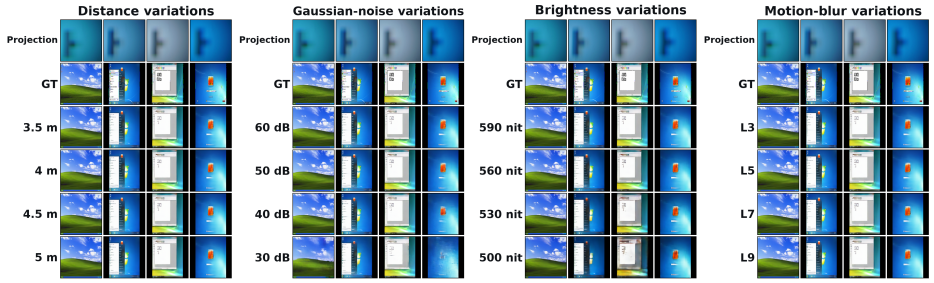

Figure 5: MCRRL recovery under controlled wall-observation variations. Distance and brightness conditions are generated by controlled image-space operators; the labels give their nominal reference values. The four side-by-side blocks show distance-related, Gaussian-noise, display-brightness, and motion-blur conditions from left to right. Each block uses the same four samples: columns identify samples, the first two rows show Input and Target, and the remaining rows show MCRRL outputs at the labeled conditions.

4.4 Robustness under Controlled Perturbations

The next question is whether screen structure remains recoverable as the wall observation changes. We hold the model weights and paired targets fixed and apply controlled image-space variations in geometry, illumination, lens response, motion, reflectance, wall response, and noise. Figure 5 shows recovered layouts and major color regions across distance-related, noise, brightness, and motion-blur conditions. The supplementary curves and examples follow these changes in greater detail, connecting the available wall cues to the screen structures reconstructed from them.

5 Conclusion

In this study, we proposed MCRRL for recovering hidden screen content from diffuse wall reflections. In MCRRL, MPRF organizes weak directional evidence into observation-conditioned structural features. Meanwhile, DRCR converts these features into foreground–background composition through a gated, clipped update. Across four real paired datasets, MCRRL leads the PSNR, SSIM, and LPIPS comparison and reconstructs brightness distributions, regional layouts, principal contours, and portions of the displayed information. Controlled changes in the wall observation further connect this recovery to the available optical evidence. Saunders et al. (2019) demonstrate ordinary-camera reconstruction of hidden two-dimensional images, including color bars, pixel-art patterns, and large letters. Using MCRRL, we extend this recovery to the internal organization of digital displays, reconstructing interface partitions, chart geometry, glyph arrangements, and browser-page blocks across four real paired datasets (Figure 3). Future work will examine how the recovered structures support text recognition and human interpretation. Together, these findings establish the preliminary feasibility of recovering hidden screen content from the light it casts onto a wall using an ordinary RGB camera.

References

Miika Aittala, Prafull Sharma, Lukas Murmann, Adam B. Yedidia, Gregory W. Wornell, William T. Freeman, and Frédo Durand. Computational mirrors: Blind inverse light transport by deep matrix factorization. In Advances in Neural Information Processing Systems, volume 32, 2019. URL https://papers.nips.cc/paper/9578-computational-mirrors-blind-inverse-light-transport-by-deep-matrix-factorization.

Saeed Anwar and Nick Barnes. Real image denoising with feature attention. In 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 3155–3164, 2019. doi: 10.1109/iccv.2019.00325.

Michael Backes, Markus Dürmuth, and Dominique Unruh. Compromising reflections-or-how to read lcd monitors around the corner. In 2008 IEEE Symposium on Security and Privacy, pp. 158–169, 2008. doi: 10.1109/SP.2008.25.

Katherine L. Bouman, Vickie Ye, Adam B. Yedidia, Frédo Durand, Gregory W. Wornell, Antonio Torralba, and William T. Freeman. Turning corners into cameras: Principles and methods. In Proceedings of the IEEE International Conference on Computer Vision, pp. 2289–2297, 2017. doi: 10.1109/ICCV.2017.249.

Hongyi Cai, Mohammad Mahdinur Rahman, Jingyu Wu, Wenzhen Dong, and Jie Li. Agileir: Memory-efficient group shifted windows attention for lightweight image restoration. In Artificial Neural Networks and Machine Learning – ICANN 2025, pp. 364–375, 2026. doi: 10.1007/978-3-032-04546-1_30. ICANN 2025; published online September 2025.

Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In European Conference on Computer Vision, Lecture Notes in Computer Science, pp. 17–33, 2022. doi: 10.1007/978-3-031-20071-7_2.

Pengyun Chen, Shuang Cui, Ning Cao, Wenhao Zhang, Pengfei Wang, Shaohui Jin, and Mingliang Xu. Lightweight multi-scale feature fusion with attention guidance for passive non-line-of-sight imaging. The Visual Computer, 41:7767–7780, 2025. doi: 10.1007/s00371-025-03837-5.

Wenzheng Chen, Simon Daneau, Colin Brosseau, and Felix Heide. Steady-state non-line-of-sight imaging. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6783–6792, 2019. doi: 10.1109/CVPR.2019.00695.

Lu Chi, Borui Jiang, and Yadong Mu. Fast fourier convolution. In Advances in Neural Information Processing Systems, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/file/2fd5d41ec6cfab47e32164d5624269b1-Paper.pdf.

Yuning Cui, Wenqi Ren, Xiaochun Cao, and Alois Knoll. Revitalizing convolutional network for image restoration. IEEE Transactions on Pattern Analysis and Machine Intelligence, 46(12):9423–9438, 2024. doi: 10.1109/TPAMI.2024.3419007.

Robinson Czajkowski and John Murray-Bruce. Two-edge-resolved three-dimensional non-line-of-sight imaging with an ordinary camera. Nature Communications, 15:1162, 2024. doi: 10.1038/s41467-024-45397-7.

Ruixu Geng, Yang Hu, Zhi Lu, Cong Yu, Houqiang Li, Hengyu Zhang, and Yan Chen. Passive non-line-of-sight imaging using optimal transport. IEEE Transactions on Image Processing, 31:110–124, 2022. doi: 10.1109/tip.2021.3128312.

Daniel Genkin, Mihir Pattani, Roei Schuster, and Eran Tromer. Synesthesia: Detecting screen content via remote acoustic side channels. In 2019 IEEE Symposium on Security and Privacy, pp. 853–869, 2019. doi: 10.1109/SP.2019.00074.

Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In Proceedings of the 27th International Conference on Machine Learning, pp. 399–406, 2010.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In Proceedings of the 34th International Conference on Machine Learning, pp. 1321–1330, 2017.

Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1712–1722, 2019. doi: 10.1109/cvpr.2019.00181.

Markus G. Kuhn. Optical time-domain eavesdropping risks of crt displays. In Proceedings of the IEEE Symposium on Security and Privacy, pp. 3–18, 2002. doi: 10.1109/SECPRI.2002.1004358.

Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, pp. 1833–1844, 2021. doi: 10.1109/ICCVW54120.2021.00210.

David B. Lindell, Gordon Wetzstein, and Matthew O'Toole. Wave-based non-line-of-sight imaging using fast f-k migration. ACM Transactions on Graphics, 38(4):1–13, 2019. doi: 10.1145/3306346.3322937.

Tomohiro Maeda, Yiqin Wang, Ramesh Raskar, and Achuta Kadambi. Thermal non-line-of-sight imaging. In 2019 IEEE International Conference on Computational Photography, pp. 1–11, 2019. doi: 10.1109/ICCPHOT.2019.8747343.

Vishal Monga, Yuelong Li, and Yonina C. Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. IEEE Signal Processing Magazine, 38(2):18–44, 2021. doi: 10.1109/MSP.2020.3016905.

Matthew O'Toole, David B. Lindell, and Gordon Wetzstein. Confocal non-line-of-sight imaging based on the light-cone transform. Nature, 555(7696):338–341, 2018. doi: 10.1038/nature25489.

Thomas Porter and Tom Duff. Compositing digital images. In Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques, pp. 253–259, 1984. doi: 10.1145/800031.808606.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention, pp. 234–241, 2015. doi: 10.1007/978-3-319-24574-4_28.

Charles Saunders, John Murray-Bruce, and Vivek K. Goyal. Computational periscopy with an ordinary digital camera. Nature, 565(7740):472–475, 2019. doi: 10.1038/s41586-018-0868-6.

Prafull Sharma, Miika Aittala, Yoav Y. Schechner, Antonio Torralba, Gregory W. Wornell, William T. Freeman, and Frédo Durand. What you can learn by staring at a blank wall. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2310–2319, 2021. doi: 10.1109/ICCV48922.2021.00233.

Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 3172–3182, 2022. doi: 10.1109/WACV51458.2022.00323.

Antonio Torralba and William T. Freeman. Accidental pinhole and pinspeck cameras: Revealing the scene outside the picture. In 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 374–381, 2012. doi: 10.1109/CVPR.2012.6247698.

Andreas Velten, Thomas Willwacher, Otkrist Gupta, Ashok Veeraraghavan, Moungi G. Bawendi, and Ramesh Raskar. Recovering three-dimensional shape around a corner using ultrafast time-of-flight imaging. Nature Communications, 3:745, 2012. doi: 10.1038/ncomms1747.

Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 17683–17693, June 2022.

Ning Xu, Brian Price, Scott Cohen, and Thomas Huang. Deep image matting. In 2017 IEEE Conference on Computer Vision and Pattern Recognition, pp. 311–320, 2017. doi: 10.1109/CVPR.2017.41.

Qingsen Yan, Yixu Feng, Cheng Zhang, Guansong Pang, Kangbiao Shi, Peng Wu, Wei Dong, Jinqiu Sun, and Yanning Zhang. Hvi: A new color space for low-light image enhancement. In 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5678–5687, 2025. doi: 10.1109/cvpr52734.2025.00533.

Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 14816–14826, 2021. doi: 10.1109/cvpr46437.2021.01458.

Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5718–5729, 2022. doi: 10.1109/CVPR52688.2022.00564.

Jian Zhang and Bernard Ghanem. Ista-net: Interpretable optimization-inspired deep network for image compressive sensing. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1828–1837, 2018. doi: 10.1109/CVPR.2018.00196.

Jiarui Zhang, Ruixu Geng, Xiaolong Du, Yan Chen, Houqiang Li, and Yang Hu. Passive non-line-of-sight imaging with light transport modulation. IEEE Transactions on Image Processing, 34:410–424, 2025. doi: 10.1109/tip.2024.3518097.

Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE Transactions on Image Processing, 26(7):3142–3155, 2017. doi: 10.1109/TIP.2017.2662206.

Kai Zhang, Yawei Li, Jingyun Liang, Jiezhang Cao, Yulun Zhang, Hao Tang, Deng-Ping Fan, Radu Timofte, and Luc Van Gool. Practical blind image denoising via swin-conv-unet and data synthesis. Machine Intelligence Research, 20:822–836, 2023. doi: 10.1007/s11633-023-1466-0.

Yu Zheng, Jiahui Zhan, Shengfeng He, Junyu Dong, and Yong Du. Curricular contrastive regularization for physics-aware single image dehazing. In 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5785–5794, 2023. doi: 10.1109/cvpr52729.2023.00560.

Shihao Zhou, Duosheng Chen, Jinshan Pan, Jinglei Shi, and Jufeng Yang. Adapt or perish: Adaptive sparse transformer with attentive feature refinement for image restoration. In 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2952–2963, 2024. doi: 10.1109/cvpr52733.2024.00285.

Supplementary Material for Architecture, Optimization, Ablations, and Robustness of MCRRL

This appendix extends the evidence chain of the main paper. The task setting, datasets, perturbation settings, detailed mechanism derivations, and extended experimental results are illustrated. The results show how varied different conditions such as brightness, distance, noise and occlusions can affect the screen recovery of MCRRL and other advanced methods from wall light spots across the reported datasets.

A Task, Data, and Evaluation

A.1 Observation Setting

We observe the wall-light pattern produced by a screen outside an ordinary RGB camera's direct view and reconstruct the corresponding screen image from that measurement. Multi-scale features derived from the observation condition the reconstruction, and paired screen targets provide supervision during training.

The data were collected in two scene configurations. Figure 1 presents one configuration, and Supplementary Figure S1 shows the other. In both configurations, the camera records wall-reflected light while the displayed content remains outside its direct view.

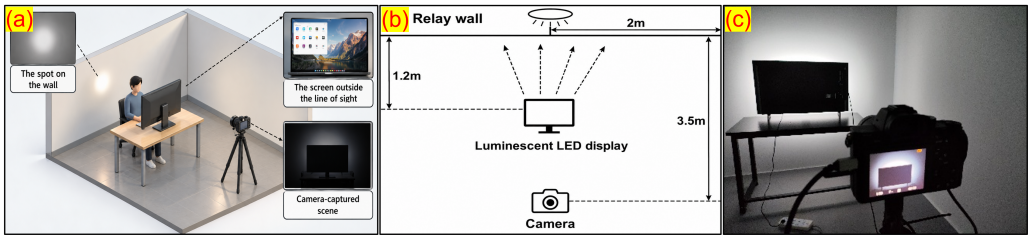

Figure S1: Second acquisition scene in our datasets. (a) Illustration of the hidden-screen setting. (b) Observation geometry. (c) Physical capture setup. The camera records the wall illumination produced by the screen while the displayed content remains outside its direct view.

A.2 Paired Datasets and Metrics

Screen, Charts, Password2, and Browser provide real paired wall spots and screen targets emphasizing interface text, chart geometry, dense glyph patterns, and browser layouts. Their complete test splits contain 374, 2,000, 1,741, and 940 pairs. Each dataset is trained independently, and image-pair metrics compare predictions with their corresponding screen targets. The evaluation measures raster fidelity, structure, edges, perceptual distance, and boundary error. PSNR, SSIM, MSE, RMSE, LPIPS, and Edge-MAE are computed for individual prediction–target pairs and then averaged; RMSE is averaged after taking each image's square root of MSE. The supplementary intervention evaluator aggregates batch-level MS-SSIM with sample-count weighting. Fréchet Inception Distance (FID; lower is better) compares the feature distributions of the reconstructed and target image sets. We use 64-dimensional Inception-v3 features through TorchMetrics Fréchet Inception Distance, accumulating predictions and targets over each complete test split after clipping to $[0, 1]$ and converting to 8-bit RGB.

Supplementary Figure S2 shows paired test examples from the four datasets.

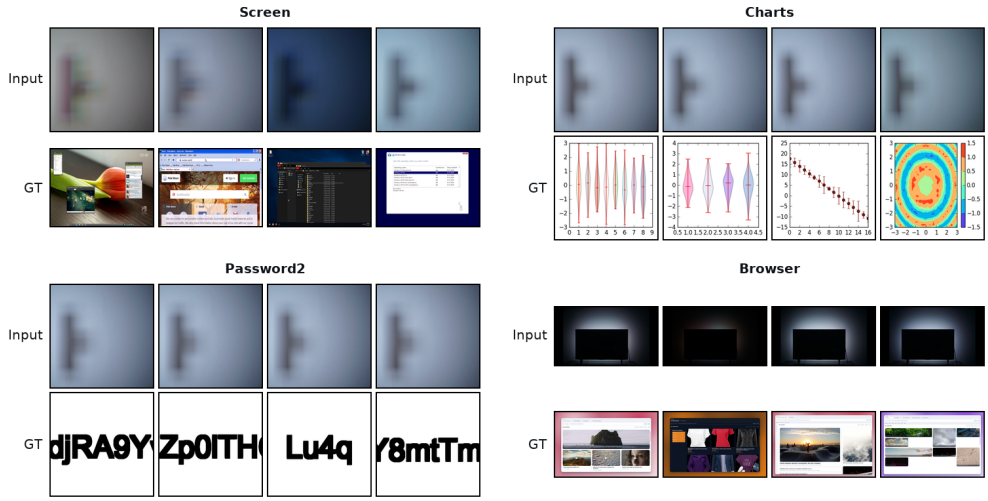

Figure S2: Paired test examples from Screen, Charts, Password2, and Browser; each block shows four RGB wall-reflection inputs (top) and corresponding screen targets (bottom). The examples preserve their source aspect ratios; evaluation uses the 256×256 grids in Table 1.

A.3 Training and Model Selection

We evaluate the complete model at the selected checkpoint for each dataset and the intervention models at their fixed training endpoints. The Screen and Browser main comparisons use clean evaluations with Lanczos-resized targets. The Screen architectural comparison retains this clean reference, whereas the Browser architectural comparison uses bicubic-resized targets. The spatial-intervention clean references use bicubic-resized targets with the same dataset-specific complete-model checkpoints and paired test samples. Each input-intervention experiment uses its corresponding clean reference.

A.4 Controlled Perturbation Protocol

Distance, view angle, position, ambient light, lens, motion, reflectance, and wall-response conditions are evaluated by applying deterministic image-space operators to the input observation while holding the target, model weights, paired split, and evaluation procedure fixed. These tests connect controlled changes in the wall observation to the fidelity, layout, and local structure of the reconstructed screen.

For Figure 5, the four columns in each block correspond to Screen samples 1657, 792, 839, and 779. Distance-related conditions use nominal distances 3.5, 4, 4.5, and 5 m, corresponding to compression factors 1, 8/7, 9/7, and 10/7 relative to 3.5 m. Display-brightness operators use nominal 590, 560, 530, and 500 nit references relative to 600 nit. Gaussian-noise levels are 60, 50, 40, and 30 dB, and motion blur uses horizontal kernels of length 3, 5, 7, and 9 pixels.

The salt-and-pepper noise sweep uses nine levels from 20 to 60 dB in steps of 5 dB. The generator relates signal power to the expected error power of random black/white replacement and uses a fixed seed of 42 plus the image index. The Gaussian examples in Figures S4–S5 add zero-mean noise with standard deviation σ to normalized RGB observations before clipping to $[0, 1]$. For the Password2 photometric-bias sweep, the reported SSIM area is the trapezoidal integral over the five ordered severity points mapped evenly to $[0, 1]$; it is normalized by that severity-index interval.

B Mechanism Derivations

B.1 Notation, Observation Abstraction, and Learning Setting

This section gives a mathematical account of MPRF and DRCR. Symbols are defined at first use, and Γ denotes an explanatory screen-to-wall light-transport operator. Unless stated otherwise, intermediate tensors are computed from the current observation in both training and inference, learned parameters are optimized during training and fixed at inference, and paired targets enter only the training objective.

B.1.1 Eq. (B.1) Paired data and normalization

We represent paired wall observations and screen targets on matching $H \times W$ image grids and normalize their values to $[0, 1]$.

$$x, y \in \mathbb{R}^{B \times C_{\mathrm{rgb}} \times H \times W}, \quad x, y \in [0, 1] \tag{B.1}$$

Here x is the RGB wall observation, y is the paired target screen image, B is the batch size, $C_{\mathrm{rgb}} = 3$ is the color-channel count, and $H \times W$ is the common image-grid resolution.

Equation (B.2) applies the convolutional projection followed by a GELU activation to the normalized observation tensor defined in Eq. (B.1).

B.1.2 Eq. (B.2) Learned input projection

The learned input projection maps the normalized observation to the initial encoder feature:

$$h_0 = \varphi(W_0 * x + d_0) \tag{B.2}$$

Here h_0 is the initial encoder feature, W_0 and d_0 are the convolution kernel and bias, φ is the elementwise GELU activation, and $*$ denotes spatial convolution.

Further structural processing is needed after the initial feature extraction in Eq. (B.2) because the paired observations exhibit attenuated fine raster structure and measurement ambiguity. The following phenomenological relation summarizes that setting.

B.1.3 Eq. (B.3) Explanatory degradation composition

We use the following phenomenological relation to summarize spatial mixing and unresolved ambiguity in the observation:

$$\tilde{x} = \Gamma(\tilde{y}) + n \tag{B.3}$$

Here $\tilde{x}$ and $\tilde{y}$ are idealized continuous observation and screen fields. The transport operator Γ describes how light from the screen is spatially mixed in the wall observation. The term n collects illumination variation, sensor noise, and model mismatch. We learn the inverse mapping from paired wall observations and screen targets.

Training fits the direct paired image-to-image mapping with the empirical objective below.

B.1.4 Eq. (B.4) Per-dataset learning objective

Each dataset is trained independently by minimizing the empirical reconstruction objective over its paired training split:

$$\theta_j^* = \arg\min_{\theta} \frac{1}{N_j} \sum_{(x,y) \in \mathcal{D}_j} \mathcal{L}\left(f_\theta(x), y\right) \tag{B.4}$$

Here $\mathcal{D}_j$ contains N_j paired samples from dataset j, f_θ is the reconstruction network, $\mathcal{L}$ is defined in Eq. (B.20), and θ_j^* is the resulting dataset-specific parameter vector.

Equation (B.4) is optimized separately for each dataset, and the conditioning path below belongs to the same dataset-specific model.

B.1.5 Eq. (B.5) Deterministic multiscale conditioning

The observation-conditioning branch applies deterministic resizing followed by a learned stem at each selected encoder level:

$$R_l = W_l^{(2)} * \varphi\Big(W_l^{(1)} * P_l(x) + d_l^{(1)}\Big) + d_l^{(2)} \tag{B.5}$$

Here P_l resizes x to level l, $W_l^{(1)}, W_l^{(2)}$ and $d_l^{(1)}, d_l^{(2)}$ are the stem parameters, and R_l is the resulting conditioning feature.

The following proximal steps use this conditioning feature to organize attenuated raster evidence.

B.2 MPRF

Indirect wall reflection attenuates directional raster evidence, so small text, thin lines, and chart elements can become weak in the observed image. Standard multiscale pathways use unconstrained dense features, allowing these cues to weaken during downsampling. Direct RGB residual regression may also average foreground and background or write unsupported texture when the measurement is ambiguous. MPRF responds by organizing oriented primitives before decoder readout.

MPRF maps an encoder feature and an observation-derived condition to a recalibrated feature on the same grid. Learned projections, masked dictionary atoms, update parameters, gates, and residual scales are combined with fixed concatenation, differencing, thresholding, pooling, and upsampling operations. The module is instantiated at selected encoder scales, with independent parameters at each scale, and passes its output to the next encoder stage or skip path.

Different encoder scales preserve different parts of the wall-light structure. We therefore instantiate MPRF with separate parameters at the selected scales, allowing each instance to organize the cues available at its resolution. Within each instance, the observation condition first establishes a reference anchor; sparse directional inference then forms a gated correction to that anchor.

The data flow within this module begins by establishing a conditional reference point for feature recalibration. The module constructs a zero-initialized-detail conditional anchor that fuses the main feature and the condition feature map through learned projections.

B.2.1 Eq. (B.6) Zero-initialized-detail conditional anchor

The feature and condition tensors are projected to matching spatial dimensions and channel width before fusion:

$$\begin{aligned} g, j &= \mathrm{split}(\Phi(R)), \\ F_m &= F \odot (1 + \tanh(g) \odot \zeta) + j \odot \zeta, \\ G_a &= \sigma(\Omega([F_m, R])), \\ A &= \Psi([F_m, R]) + \Delta([R, L_h(R), L_v(R)]) \odot G_a \odot \zeta. \end{aligned} \tag{B.6}$$

The main feature F comes from the preceding encoder block and the condition R from the resize stem. The projection Φ produces modulation maps g and j, ζ is a zero-initialized per-channel scale, and L_h, L_v are depthwise horizontal and vertical filters. The learned maps

Ψ, Δ, and Ω produce the baseline projection, line-detail projection, and sigmoid gate G_a, yielding the conditional anchor A.

The module next forms a measurement descriptor from the feature, the condition, and their elementwise absolute difference.

B.2.2 Eq. (B.7) Measurement descriptor

The descriptor concatenates the two features and their absolute difference before a learned projection:

$$M = W_d^{(2)} * \varphi\Big(W_d^{(1)} * [F, R, |F - R|] + b_d^{(1)}\Big) + b_d^{(2)}, \qquad M_p = P(M) \tag{B.7}$$

Here $W_d^{(1)}, W_d^{(2)}$ and $b_d^{(1)}, b_d^{(2)}$ parameterize the descriptor projection, φ is GELU, and P is fixed average pooling. The full-resolution descriptor M and pooled descriptor M_p therefore occupy different grids; only M_p enters the sparse objective.

The module then applies a finite unrolled update associated with a sparse surrogate objective.

B.2.3 Eq. (B.8) Sparse explanation surrogate

The pooled descriptor is represented with an orientation-constrained learned dictionary and a sparsity penalty:

$$\min_Z \frac{1}{2}||DZ - M_p||_2^2 + \lambda_k||Z||_1 \tag{B.8}$$

Here Z is the coefficient tensor, D is the learned dictionary, and $\lambda_k > 0$ controls sparsity at step k. Both norms reduce over channel and spatial dimensions, producing a scalar objective per sample and fusion site. The step-specific sparsity levels are learned jointly with the dictionary.

The dictionary uses fixed orientation masks and atom normalization. The masks group atoms into horizontal, vertical, diagonal, and free-form supports, providing an architectural prior for raster structure.

B.2.4 Eq. (B.9) Orientation-masked normalized dictionary

Fixed masks divide dictionary atoms into horizontal, vertical, diagonal, and free-form support classes, followed by per-atom normalization:

$$D_a = \frac{M_o \odot V_a}{\max(||M_o \odot V_a||_2, \epsilon)} \tag{B.9}$$

For atom a, V_a is the unconstrained kernel, M_o is its assigned binary orientation mask, and $\epsilon > 0$ prevents division by zero. The norm spans descriptor channels and spatial support, so every masked atom above the normalization floor has unit norm; D collects all resulting atoms D_a.

The recursion starts from a zero code and applies K unrolled proximal steps. Each step composes a gradient update on the quadratic explanation term with the proximal operator of the sparsity penalty, and the dictionary adjoint appears because the analysis operator is the transpose of the synthesis operator.

B.2.5 Eq. (B.10) Unrolled proximal update with soft thresholding

Starting from a zero code, each unrolled step applies a gradient update followed by soft thresholding:

$$\bar{Z}^{k+1} = \mathcal{S}_{\theta_k}(Z^k - \eta_k D^T(DZ^k - M_p)), \quad \theta_k = \eta_k \lambda_k \tag{B.10}$$

Here $k = 0, \ldots, K-1$, $\bar{Z}^{k+1}$ denotes the proximal update before consensus, D and D^T are synthesis and analysis mappings, and η_k is a bounded learned step size. The threshold $\theta_k = \eta_k \lambda_k$ sets coefficients below θ_k in magnitude to zero and shrinks the remainder toward zero.

The pooled-and-upsampled consensus term mixes each iterate with the difference between a coarser copy and the current code. Because the learned bounded coefficient is signed, the update can move toward or away from that copy.

B.2.6 Eq. (B.11) Coarse-to-fine consensus step

Pooling and upsampling form a coarse copy whose contribution is controlled by a bounded learned coefficient:

$$Z^{k+1} = \bar{Z}^{k+1} + \rho_k(U_c(P_c(\bar{Z}^{k+1})) - \bar{Z}^{k+1}), \qquad \rho_k = \rho_m \tanh(a_k) \tag{B.11}$$

The operators P_c and U_c act on the coefficient grid and are distinct from the image resize P_l in Eq. (B.5). The learned logit a_k produces $\rho_k \in (-\rho_m, \rho_m)$, and the consensus-corrected final state Z^K is passed to Eqs. (B.12)–(B.13).

After the unrolled steps, the module compares the explanation energies before and after the update and maps their signed relative difference through a learned sigmoid gate. The formulation allows negative differences when $E_K > E_0$.

B.2.7 Eq. (B.12) Signed energy-difference gate

The signed change in local explanation energy is mapped to a spatial gate:

$$G = \sigma\left(\text{softplus}(\gamma)\frac{E_0 - E_K}{\text{clamp}_{\min}(E_0, \epsilon)} + \beta\right) \tag{B.12}$$

Here $E_0 = \text{mean}_c(M_p^2)$ and $E_K = \text{mean}_c((DZ^K - M_p)^2)$ are channel-averaged energy maps on the pooled grid. Their signed relative difference is scaled by learned γ, β, stabilized by $\epsilon > 0$, and mapped by sigmoid to $G \in (0, 1)$.

The injection multiplies the synthesized update by the learned gate and a zero-initialized scale before adding it to the anchor, making the update exactly zero at initialization.

B.2.8 Eq. (B.13) Zero-initialized primitive injection

The synthesized code is projected as a spatial correction, with a zero-initialized scale that returns the conditional anchor at initialization:

$$F_o = A + \Pi\left(U_p(DZ^K \odot G)\right) \odot \tanh(\xi) \tag{B.13}$$

The final code Z^K is synthesized by D, gated by G, resampled by bilinear operator U_p, and mapped by Π to the anchor width. The per-channel parameter ξ is initialized at zero and enters through $\tanh(\xi)$, producing the output feature F_o on the anchor grid.

B.3 DRCR

The final decoder operates under measurement ambiguity, where a direct continuous residual can mix color regions or add texture in flat areas. This motivates a bounded two-field composition as an architectural prior.

The DRCR module operates at the final decoder scale. It receives the final high-resolution decoder feature, a coarse residual, and the original observation; learned heads produce a latent opacity field, two color fields, and a bounded gate. Fixed sigmoid, convex-combination, channel-mean absolute-difference, and symmetric-clipping operations form a composition residual, which is added with the observation and a parallel learned RGB correction. The active design uses the normalized image grid and a learned red, green, and blue output scale.

The data flow within this module begins by encoding the decoder feature together with the observation and coarse reconstruction state, then projecting the resulting latent feature into compositional fields.

B.3.1 Eq. (B.14) Latent opacity and layer fields

The latent encoder combines the decoder feature, observation, and coarse reconstruction, then predicts bounded compositional fields:

$$u = e([H_d, x + r_0, x]), \quad [\ell_\alpha, \ell_f, \ell_b] = W_p * u + b_p, \quad \alpha = \sigma(\ell_\alpha), \quad f = \sigma(\ell_f), \quad b = \sigma(\ell_b) \tag{B.14}$$

The encoder e maps the concatenated inputs to u. Projection parameters W_p, b_p produce opacity logit ℓ_α and color logits ℓ_f, ℓ_b; sigmoid maps them to $\alpha \in [0,1]^{B\times 1\times H\times W}$ and $f, b \in [0,1]^{B\times C_{\text{rgb}}\times H\times W}$.

The composition stays inside the unit interval because it is a convex combination of two fields that are themselves squashed into the unit interval by Eq. (B.14).

B.3.2 Eq. (B.15) Convex digital composition

The latent fields form a convex digital composition:

$$c = \alpha \odot f + (1 - \alpha) \odot b \tag{B.15}$$

The single-channel opacity α is broadcast across color channels, $\odot$ denotes elementwise multiplication, and $c \in [0,1]^{B\times C_{\text{rgb}}\times H\times W}$ is the resulting composition. The fields are latent digital layers that provide a structural prior for the readout.

To assess the distinctness of the predicted layers, the module computes a spatial map that averages the absolute difference between the foreground and background color fields derived in Eq. (B.14) and Eq. (B.15).

B.3.3 Eq. (B.16) Layer separation map

The channel-mean absolute difference between the two latent color fields defines the separation feature:

$$s = \frac{1}{3}\sum_{j=1}^{3} |f_j - b_j| \tag{B.16}$$

Here $C_{\text{rgb}} = 3$, j indexes RGB channels, and $s \in \mathbb{R}^{B\times 1\times H\times W}$. The separation map measures the local color difference between the predicted foreground and background. We combine this map with the latent feature to learn the composition gate through paired screen-reconstruction supervision.

The layer separation map from Eq. (B.16) is then combined with the latent feature to form the bounded learned gate used by the writeback.

B.3.4 Eq. (B.17) Learned composition gate

The gate head combines the latent feature and separation map to produce a bounded spatial score:

$$q = \sigma\Big(W_q^{(2)} * \varphi\Big(W_q^{(1)} * [u, s] + b_q^{(1)}\Big) + b_q^{(2)}\Big) \tag{B.17}$$

The features u and s are concatenated along channels; $W_q^{(1)}, W_q^{(2)}$ and $b_q^{(1)}, b_q^{(2)}$ parameterize the gate head, φ is GELU, and sigmoid yields $q \in (0,1)^{B\times 1\times H\times W}$. Paired reconstruction supervision learns the gate as a latent reliability field.

The writeback first clips the composition difference, then multiplies it by the learned gate from Eq. (B.17) and an unconstrained learned RGB scale.

B.3.5 Eq. (B.18) Clipped and scaled composition writeback

Clipping controls the composition difference before spatial gating and RGB scaling.

$$r_c = r_0 + \mathrm{clip}(c - (x + r_0), -\tau, \tau) \odot q \odot v \tag{B.18}$$

The residual r_c updates the coarse residual r_0 using the difference between the composition c and the current coarse reconstruction. The fixed positive constant τ is the symmetric clip bound, q is broadcast across color channels, and v is a learned per-channel output scale. The resulting update satisfies $|r_c - r_0| \le \tau|v|$ elementwise, linking its magnitude directly to the clip threshold and the learned RGB scale.

The final network prediction aggregates all residual components to form the reconstructed screen image, combining the clipped and scaled writeback from Eq. (B.18) with the parallel learned RGB correction.

B.3.6 Eq. (B.19) Aggregated network readout

The final readout adds the composition-refined residual and parallel RGB correction to the observation:

$$\hat{y} = x + r_c + r_f \tag{B.19}$$

Here $\hat{y}$ is the reconstructed screen tensor, x is the observation, r_c is the composition-refined residual, and r_f is the parallel learned RGB correction for effects outside the composition path.

Paired reconstruction supervision alone learns the opacity and composition gate as latent fields.

B.4 Support Components, Training Objective, and Inference Readout

The architecture also incorporates a parallel learned RGB correction head and a composite objective. The residual head supplies an additional RGB correction, while the objective combines pixel, structural, edge, perceptual, and feature-statistics terms.

The parallel correction head combines the final decoder feature H_d with an input-gradient map:

$$g_x = \mathrm{clip}\left(\frac{m_x}{3(\langle m_x\rangle_{hw} + 10^{-6})}, 0, 1\right), \quad m_x = \sqrt{(\partial_h \bar{x})^2 + (\partial_v \bar{x})^2 + 10^{-6}},$$
$$h_f = P_f(H_d), \quad h_g = P_g(g_x), \quad r_f = T(h_f + h_g) \odot \sigma(Q_f([h_f, h_g])) \odot v_f.$$

Here $\bar{x}$ is mean RGB intensity. We first take forward differences and then replicate the last row or column of each difference tensor at the corresponding boundary; this preserves the implemented finite-difference convention rather than padding the input before differencing.

The learned 3×3 projections P_f, P_g, each followed by GELU, feed a 3×3 depthwise convolution, GELU, a dilated 5×5 depthwise convolution with dilation 2, GELU, and an RGB projection in T; the two-layer 1×1 gate Q_f spatially weights this correction, with learned RGB scale v_f initialized to 0.1.

B.4.1 Eq. (B.20) Training objective with scheduled support terms

Training uses paired targets and a fixed differentiable perceptual embedding. Positive coefficients weight the pixel, structural, edge, perceptual, and feature-statistics terms, while a bounded progress function introduces the support terms during optimization.

$$\mathcal{L} = \mathcal{L}_x + \lambda_s \mathcal{L}_s + \lambda_e \mathcal{L}_e + \varrho(t)(\lambda_v \mathcal{L}_v + \lambda_f \mathcal{L}_f) \tag{B.20}$$

The weights are $(\lambda_s, \lambda_e, \lambda_v, \lambda_f) = (0.2, 0.1, 0.020, 0.010)$, and the support ramp is $\varrho(t) = \mathrm{clip}((t - 0.20)/0.40, 0, 1)$ for epoch fraction t. Writing $\langle \cdot \rangle$ for the mean over tensor entries, the losses are

$$\begin{aligned}
\mathcal{L}_x &= \langle |\hat{y} - y| \rangle, \\
\mathcal{L}_s &= \mathrm{clip}\left(\frac{1 - \langle S(\hat{y}_c, y) \rangle}{2}, 0, 1 \right), \\
\mathcal{L}_e &= \langle |\partial_h \hat{y}_c - \partial_h y| \rangle + \langle |\partial_v \hat{y}_c - \partial_v y| \rangle, \\
\mathcal{L}_v &= \mathrm{LPIPS}_{\mathrm{AlexNet}}(2\mathcal{R}_{128}(\hat{y}_c) - 1, 2\mathcal{R}_{128}(y) - 1), \\
\mathcal{L}_f &= \tfrac{1}{2} \sum_{a \in \{1/2, 1/4\}} \left(\langle |m_a(\hat{y}_c) - m_a(y)| \rangle + 0.25 \langle |K_a(\hat{y}_c) - K_a(y)| \rangle \right).
\end{aligned}$$

The local structural score is

$$S(p, y) = \frac{(2\mu_p \mu_y + 0.01^2)(2\sigma_{py} + 0.03^2)}{(\mu_p^2 + \mu_y^2 + 0.01^2)(\sigma_p^2 + \sigma_y^2 + 0.03^2) + 10^{-6}},$$

where moments use a 7×7 mean window with zero padding. The finite differences ∂_h, ∂_v use the forward-difference-then-last-row/column-replication convention defined above. LPIPS uses frozen AlexNet features and bilinear resizing to 128×128 for larger images; its value is reduced by an arithmetic mean over batch samples when a batch is evaluated. For the statistics term, area resizing forms each scale; its feature stack contains RGB, horizontal and vertical grayscale differences, the grayscale high-pass residual after a half-scale down/up resize, and absolute grayscale deviation from a 7×7 local mean. For feature matrix $F_a \in \mathbb{R}^{d \times N_a}$, $m_a = N_a^{-1} F_a \mathbf{1}$ and $K_a = (F_a - m_a \mathbf{1}^T)(F_a - m_a \mathbf{1}^T)^T / (N_a - 1)$. The pixel loss uses the continuous prediction, whereas all four complementary terms use $\hat{y}_c$.

After the learned forward path, the system applies the same deterministic clamping operation before computing the structural, edge, perceptual, and feature-statistics losses and evaluation.

B.4.2 Eq. (B.21) Normalized Prediction for Training and Evaluation

The structural, edge, perceptual, and feature-statistics losses and evaluation clip the additive readout to the normalized range $[0, 1]$:

$$\hat{y}_c = \mathrm{clip}(\hat{y}, 0, 1) = \mathrm{clip}(x + r_c + r_f, 0, 1) \tag{B.21}$$

The continuous prediction $\hat{y}$ supplies the pixel loss. Its projection $\hat{y}_c$ supplies the structural, edge, perceptual, and feature-statistics losses and the evaluation metrics.

B.5 Diagnostic Definitions Used by the Extended Tables

To evaluate the stability of the proposed method under various degradation conditions, the extended tables rely on specific diagnostic definitions that quantify performance retention and spatial sensitivity. These metrics isolate the impact of controlled image-space perturba-

tions, such as geometry shifts, material variations, and additive noise, on the reconstruction fidelity.

B.5.1 Eq. (B.22) Direction-normalized score ratio used by the perturbation summaries

The reported clean and perturbed metric scores are positive. The piecewise ratio is a metric-specific descriptive summary, especially for dB-valued metrics.

$$\mathcal{V} = \begin{cases} \mu_p/\mu_c, & \nu = +1, \\ \mu_c/\mu_p, & \nu = -1. \end{cases} \tag{B.22}$$

Here μ_p and μ_c are perturbed and clean metric scores, while $\nu = +1$ for higher-is-better metrics and $\nu = -1$ for lower-is-better metrics. For PSNR, the ratio defines a scale-dependent comparison of decibel values. We call this quantity the PSNR score ratio. Figure S34 instead shows relative LPIPS increase, $I_{\mathrm{LPIPS}} = 100(\mu_p - \mu_c)/\mu_c$, alongside $100\mu_p/\mu_c$ for PSNR and SSIM.

We align score differences with each metric's optimization direction.

B.5.2 Eq. (B.23) Direction-normalized trained model score difference

The difference is expressed in the units of the corresponding metric.

$$\Delta_{\mathrm{score}} = \nu \cdot (\mu_f - \mu_a) \tag{B.23}$$

Here μ_f and μ_a are the complete-model and intervention-model scores. With the direction multiplier ν, positive Δ_{score} indicates an advantage for the complete model.

Separately, the spatial diagnostic partitions the input into a three-by-three grid and masks or retains each region to report the fixed model's response.

B.5.3 Eq. (B.24) Nine-grid spatial sensitivity probe

The observation grid is partitioned into nine regions. Interventions replace the removed support with a constant RGB field equal to the spatial mean of the current observation.

$$x_m^+ = x \odot G_m + a \odot (1 - G_m), \qquad x_m^- = x \odot (1 - G_m) + a \odot G_m \tag{B.24}$$

Here G_m is the binary mask for sector $m \in \{1, \ldots, 9\}$, and a broadcasts the input's per-channel spatial mean across the image. Thus x_m^+ retains only sector m against that mean field, and x_m^- replaces that sector. The sector index is independent of the proximal-step index in Eq. (B.10).

C Extended Experimental Results

This section reports the extended metric, robustness, and diagnostic results.

C.1 Extended Comparisons on Four Datasets

The extended comparison complements the compact main table with multiscale structure, pixel error, and edge error. Supplementary Tables S1–S4 preserve the dataset-specific method pools and paired test splits.

Table S1: Extended metric comparison on Screen. MS-SSIM is higher is better; MSE, RMSE, and Edge-MAE are lower is better. Red and blue mark the best and second-best values within the dataset–metric group, and row fills identify rows containing marked values.

Method	MS-SSIM ↑	MSE ↓	RMSE ↓	Edge-MAE ↓
MCRRL (full model)	0.844583	0.00804786	0.070068	0.0311164
AST	0.733127	0.0124	0.0957648	0.0344264
Uformer	0.713958	0.0124947	0.0988577	0.0364925
NAFNet	0.715528	0.0149237	0.102263	0.0377364
ConvIR	0.695935	0.0143744	0.102963	0.0364235
HVI-CIDNet	0.67619	0.0164185	0.110406	0.0361326
SCUNet	0.679148	0.0168545	0.113522	0.0367298
LMS-NLOS	0.684939	0.0141587	0.105233	0.0390152
U-Net	0.685875	0.0186253	0.118307	0.037084
MPRNet	0.636502	0.0196609	0.122993	0.0358802
RIDNet	0.549226	0.0322986	0.157542	0.0388655
CBDNet	0.56257	0.032448	0.157152	0.0402169
DRS	0.564992	0.0293044	0.154624	0.0390651
NLOST	0.621015	0.0331296	0.1696	0.0369942
C2PNet	0.500248	0.0400334	0.180354	0.041329
DnCNN	0.447506	0.0500534	0.203819	0.0433464
SwinIR	0.453426	0.0531877	0.20917	0.041982
AgileIR	0.423837	0.0548728	0.212237	0.0449121
NLOSNeuS	0.358224	0.0903764	0.272184	0.0426464
NLOSOT	0.278022	0.137443	0.344148	0.0407466

Table S2: Extended metric comparison on Charts; metric directions and highlighting conventions follow Supplementary Table S1.

Method	MS-SSIM ↑	MSE ↓	RMSE ↓	Edge-MAE ↓
AST	0.621361	0.0245074	0.148685	0.045305
MCRRL (full model)	0.638308	0.02463	0.148053	0.0477581
Uformer	0.606932	0.0246654	0.149259	0.0457594
ConvIR	0.581909	0.0265268	0.154327	0.0461769
NAFNet	0.574185	0.0276647	0.156997	0.0488012
HVI-CIDNet	0.54844	0.0293829	0.162335	0.0469774
LMS-NLOS	0.576631	0.0267249	0.155188	0.0496259
MPRNet	0.538593	0.0298183	0.163905	0.0464246
U-Net	0.564173	0.0296524	0.163489	0.0484629
SCUNet	0.531088	0.0324197	0.170964	0.0478707
DRS	0.50891	0.0341209	0.175698	0.0476127
C2PNet	0.473677	0.036531	0.181295	0.0477504
RIDNet	0.474147	0.0376191	0.183678	0.0490159
CBDNet	0.469187	0.0387902	0.185823	0.0502539
NLOST	0.456231	0.0411352	0.189643	0.0469264
SwinIR	0.416777	0.0465813	0.202371	0.0477529
AgileIR	0.36035	0.0732034	0.240003	0.0462797
DnCNN	0.435092	0.0434362	0.195809	0.0488365
NLOSNeuS	0.359461	0.0744184	0.243616	0.0463849
NLOSOT	0.190281	0.085728	0.277168	0.0579935

Table S3: Extended metric comparison on Password2; metric directions and highlighting conventions follow Supplementary Table S1.

Method	MS-SSIM ↑	MSE ↓	RMSE ↓	Edge-MAE ↓
MCRRL (full model)	0.93465	0.0232792	0.146968	0.0274386
Uformer	0.88724	0.0371775	0.190355	0.0345503
AST	0.869055	0.03761	0.191668	0.0337176
ConvIR	0.862466	0.0413788	0.201835	0.0357343
MPRNet	0.840903	0.0479243	0.217408	0.0332037
LMS-NLOS	0.840188	0.0480397	0.217581	0.0397995
NAFNet	0.824869	0.050746	0.223639	0.0363349
HVI-CIDNet	0.778398	0.064051	0.251708	0.0309576
DRS	0.79192	0.0646847	0.253075	0.0344121
U-Net	0.796866	0.0575291	0.239046	0.0440014
CBDNet	0.719106	0.0829421	0.286578	0.0328275
NLOST	0.746634	0.0651148	0.253723	0.0319426
C2PNet	0.711934	0.0779073	0.277859	0.0337809
SwinIR	0.600644	0.11278	0.334508	0.027069
AgileIR	0.480802	0.132454	0.362689	0.0233057
DnCNN	0.655017	0.0870768	0.294202	0.0406581
RIDNet	0.48168	0.132862	0.363237	0.0260052
SCUNet	0.479786	0.134708	0.365769	0.0229775
NLOSNeuS	0.479553	0.134646	0.365685	0.0234178
NLOSOT	0.489652	0.134003	0.365427	0.0263154

Table S4: Extended metric comparison on Browser; metric directions and highlighting conventions follow Supplementary Table S1.

Method	MS-SSIM ↑	MSE ↓	RMSE ↓	Edge-MAE ↓
MCRRL (full model)	0.650947	0.0236587	0.138498	0.0438692
Uformer	0.630338	0.0240311	0.1413	0.0414214
AST	0.620111	0.0259945	0.146633	0.0416366
NAFNet	0.613937	0.0273693	0.149079	0.0447351
ConvIR	0.61802	0.0253355	0.145002	0.0411646
NLOSOT	0.61351	0.0275491	0.149439	0.0410301
LMS-NLOS	0.604502	0.0263643	0.147304	0.0457445
SCUNet	0.588485	0.0297629	0.157364	0.0430139
Restormer	0.564421	0.0326773	0.163275	0.0437794
MPRNet	0.581972	0.0280003	0.153506	0.0416352
HVI-CIDNet	0.571545	0.0317501	0.160896	0.0428656
RIDNet	0.543644	0.03286	0.166319	0.0452259
C2PNet	0.520739	0.0362889	0.173798	0.0462353
DRS	0.519798	0.03814	0.178244	0.0465633
NLOST	0.558352	0.0333815	0.170788	0.0464147
SwinIR	0.385887	0.0605563	0.227834	0.0535054
U-Net	0.401032	0.0713579	0.245264	0.0513321
DnCNN	0.375347	0.0768813	0.254537	0.0558941
AgileIR	0.353358	0.073302	0.24982	0.0602274
CBDNet	0.302307	0.313646	0.506511	0.0421612
NLOSNeuS	0.179165	0.141998	0.373507	0.0543912

The structural advantage extends across scales. MCRRL achieves the highest MS-SSIM on all four datasets, complementing the primary-metric comparison with evidence about multiscale image organization. This result is consistent with the recovered regional layouts and local structures in Figure 3 and with the multiscale representation used by MPRF. Cui et al. (2024) use a convolutional architecture in ConvIR for image dehazing, deblurring, deraining, and desnowing. In our Screen comparison (Table S1), MCRRL reduces MSE from 0.0143744 to 0.00804786 and Edge-MAE from 0.0364235 to 0.0311164 relative to ConvIR, improving both pixel agreement and edge fidelity.

Tables S5–S8 extend the ablation study from the two principal modules to their fusion, scale placement, conditioning, and reconstruction components.

Selective-Kernel Fusion and Bidirectional Pyramid replace the MPRF fusion block at the same interface; Residual Dense-Block and Activation-Free Readout replace the DRCR reconstruction head. Final-Step Consensus applies coefficient consensus only after the final proximal step; Pooling Factor 4 changes descriptor pooling from two to four. Finest-Scale-Only, Two-Coarser-Scales, and Coarsest-Scale-Only activate MPRF at 1/2, at 1/4 and 1/8, or at 1/8 resolution, respectively. Topology-Conditioned Head replaces composition with the topology-conditioned reconstruction mode. Auxiliary Readout replaces DRCR with the axis, detail, and line-adapter reconstruction paths, whereas Added Auxiliary Readout retains DRCR and adds those paths. Alternative Conditioning replaces MPRF with task fusion. Its Topology variant uses the topology-conditioned head, while its Auxiliary Readout variant retains DRCR and adds the auxiliary reconstruction paths.

Table S5: Additional network replacements and internal-structure ablations on Screen. PSNR and SSIM are higher is better; LPIPS is lower is better. Red and blue indicate the best and second-best values within each metric column.

Variant	PSNR ↑	SSIM ↑	LPIPS ↓
Full MCRRL	24.986	0.8242	0.2328
MPRF w/ Selective-Kernel Fusion	23.074	0.7714	0.3052
MPRF w/ Bidirectional Pyramid	22.67	0.7607	0.3198
DRCR w/ Residual Dense-Block Readout	23.208	0.7708	0.2960
DRCR w/ Activation-Free Readout	23.479	0.7877	0.2898
MPRF w/o Coefficient Consensus	23.478	0.7895	0.2914
MPRF with Final-Step Consensus	23.385	0.7899	0.2919
MPRF with Single-Step Solver	23.359	0.7841	0.2905
MPRF with Pooling Factor 4	23.47	0.7872	0.2909
MPRF with Finest-Scale-Only	23.43	0.7841	0.2936
MPRF with Two-Coarser-Scales	23.313	0.7814	0.2946
MPRF with Coarsest-Scale-Only	23.224	0.7735	0.2968
DRCR with Fixed Opacity	23.485	0.7898	0.2905
DRCR with Direct Foreground	23.379	0.7888	0.2918
DRCR with Unit Gate	23.432	0.7873	0.2916
DRCR with Topology-Conditioned Head	23.419	0.7887	0.2918
DRCR replaced by Auxiliary Readout	23.351	0.7767	0.2891
DRCR with Added Auxiliary Readout	23.469	0.7824	0.2922
DRCR with Alternative Conditioning + Topology	23.29	0.7796	0.2981
DRCR with Alternative Conditioning + Auxiliary Readout	22.979	0.7732	0.3039

Table S6: Additional network replacements and internal-structure ablations on Charts; evaluation and highlighting conventions follow Supplementary Table S5.

Variant	PSNR ↑	SSIM ↑	LPIPS ↓
Full MCRRL	17.14	0.7298	0.4422
MPRF w/ Selective-Kernel Fusion	15.816	0.6889	0.6106
MPRF w/ Bidirectional Pyramid	15.474	0.6864	0.6267
DRCR w/ Residual Dense-Block Readout	16.795	0.7184	0.5436
DRCR w/ Activation-Free Readout	16.794	0.7186	0.5369
MPRF w/o Coefficient Consensus	17.016	0.7208	0.4700
MPRF with Final-Step Consensus	16.93	0.7194	0.4720
MPRF with Single-Step Solver	16.992	0.7224	0.4694
MPRF with Pooling Factor 4	17	0.7220	0.4681
MPRF with Finest-Scale-Only	16.974	0.7237	0.4769
MPRF with Two-Coarser-Scales	16.942	0.7210	0.4712
MPRF with Coarsest-Scale-Only	16.944	0.7187	0.4730
DRCR with Fixed Opacity	16.951	0.7219	0.4703
DRCR with Direct Foreground	16.927	0.7204	0.4688
DRCR with Unit Gate	16.921	0.7182	0.4724
DRCR with Topology-Conditioned Head	16.961	0.7213	0.4711
DRCR replaced by Auxiliary Readout	16.936	0.7184	0.4737
DRCR with Added Auxiliary Readout	16.899	0.7200	0.4695
DRCR with Alternative Conditioning + Topology	16.585	0.7144	0.5252
DRCR with Alternative Conditioning + Auxiliary Readout	16.889	0.7219	0.4698

Table S7: Additional network replacements and internal-structure ablations on Password2; evaluation and highlighting conventions follow Supplementary Table S5.

Variant	PSNR ↑	SSIM ↑	LPIPS ↓
Full MCRRL	17.06	0.9183	0.0719
MPRF w/ Selective-Kernel Fusion	15.148	0.8901	0.1146
MPRF w/ Bidirectional Pyramid	11.408	0.8015	0.2933
DRCR w/ Residual Dense-Block Readout	15.368	0.8923	0.1085
DRCR w/ Activation-Free Readout	15.324	0.8927	0.1067
MPRF w/o Coefficient Consensus	16.246	0.9082	0.0847
MPRF with Final-Step Consensus	16.476	0.9105	0.0803
MPRF with Single-Step Solver	16.443	0.9111	0.0804
MPRF with Pooling Factor 4	16.213	0.9087	0.0842
MPRF with Finest-Scale-Only	16.289	0.9077	0.0836
MPRF with Two-Coarser-Scales	16.117	0.9047	0.0871
MPRF with Coarsest-Scale-Only	16.293	0.9066	0.0834
DRCR with Fixed Opacity	16.346	0.9088	0.0831
DRCR with Direct Foreground	15.721	0.9040	0.0918
DRCR with Unit Gate	16.078	0.9061	0.0836
DRCR with Topology-Conditioned Head	16.471	0.9099	0.0814
DRCR replaced by Auxiliary Readout	16.026	0.9010	0.0853
DRCR with Added Auxiliary Readout	16.14	0.9053	0.0865
DRCR with Alternative Conditioning + Topology	16.397	0.9104	0.0823
DRCR with Alternative Conditioning + Auxiliary Readout	16.172	0.9067	0.0869

Table S8: Additional network replacements and internal-structure ablations on Browser; evaluation and highlighting conventions follow Supplementary Table S5.

Variant	PSNR ↑	SSIM ↑	LPIPS ↓
Full MCRRL	18.353	0.6733	0.3447
MPRF w/ Selective-Kernel Fusion	17.178	0.6395	0.4992
MPRF w/ Bidirectional Pyramid	16.735	0.6264	0.5165
DRCR w/ Residual Dense-Block Readout	17.105	0.6407	0.4986
DRCR w/ Activation-Free Readout	17.331	0.6441	0.4907
MPRF w/o Coefficient Consensus	17.956	0.6688	0.3848
MPRF with Final-Step Consensus	17.853	0.6640	0.3835
MPRF with Single-Step Solver	18.001	0.6676	0.3814
MPRF with Pooling Factor 4	18.045	0.6677	0.3773
MPRF with Finest-Scale-Only	17.943	0.6688	0.3839
MPRF with Two-Coarser-Scales	18.026	0.6705	0.3756
MPRF with Coarsest-Scale-Only	18.055	0.6707	0.3797
DRCR with Fixed Opacity	18.137	0.6698	0.3762
DRCR with Direct Foreground	18.041	0.6697	0.3803
DRCR with Unit Gate	18.019	0.6681	0.3821
DRCR with Topology-Conditioned Head	18.074	0.6680	0.3773
DRCR replaced by Auxiliary Readout	18.141	0.6683	0.3756
DRCR with Added Auxiliary Readout	18.026	0.6685	0.3780
DRCR with Alternative Conditioning + Topology	17.468	0.6559	0.4188
DRCR with Alternative Conditioning + Auxiliary Readout	18.062	0.6687	0.3822

C.2 Additive Noise and Photometric Bias

Weak screen cues must remain useful when the wall image is corrupted. We first examine two distinct changes: noise perturbs local intensity patterns, while photometric bias shifts the brightness reference for foreground–background reconstruction. Figures S3–S7 follow

how these changes affect recovered layouts and glyph structures, linking the perturbation curves to visible screen information.

Under the mildest salt-and-pepper condition at 60 decibels, the proposed system achieves an SSIM of 0.82766.

On Password2 under photometric bias, the normalized SSIM area under the severity-index curve is 0.91436 and the worst-case SSIM is 0.90846.

Supplementary Figure S3 plots the Screen response to salt-and-pepper noise over the nine evaluated levels from 20 to 60 decibels.

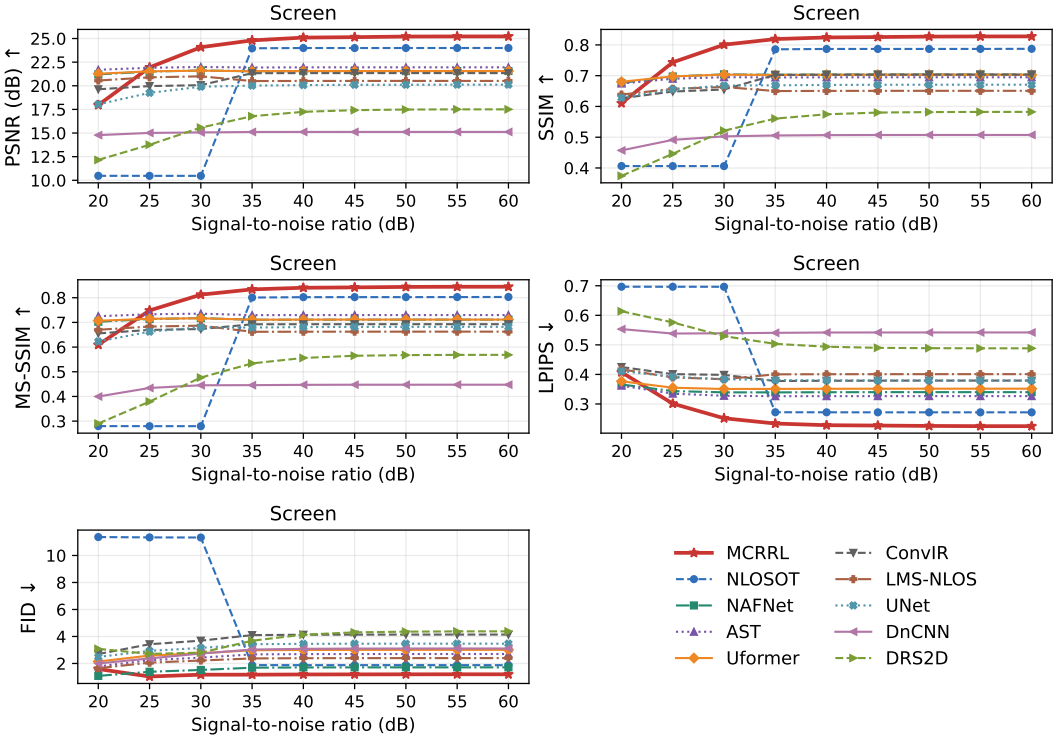

Figure S3: Screen reconstruction under salt-and-pepper noise. The five panels show PSNR, SSIM, MS-SSIM, LPIPS, and FID over nine noise levels from 20 to 60 dB.

Supplementary Figure S4 shows how noise grain, structural outlines, and error details evolve as the Gaussian operator strengthens.

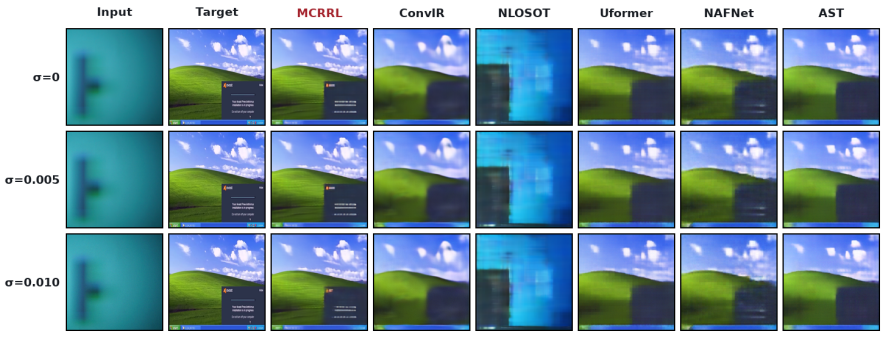

Figure S4: One Screen example with Gaussian-noise levels $\sigma = 0$, 0.005, and 0.010.

The main screen layout remains recoverable as local noise is added to the wall observation. Across the three levels, MCRRL preserves the sky–ground partition, foreground dialog, and

bottom taskbar. These structures occupy different parts of the target screen, showing that the reconstruction retains their overall arrangement while using the corrupted wall evidence.

Supplementary Figure S5 extends the noise analysis to Password2 and shows how increasing input grain changes high-contrast stroke boundaries.

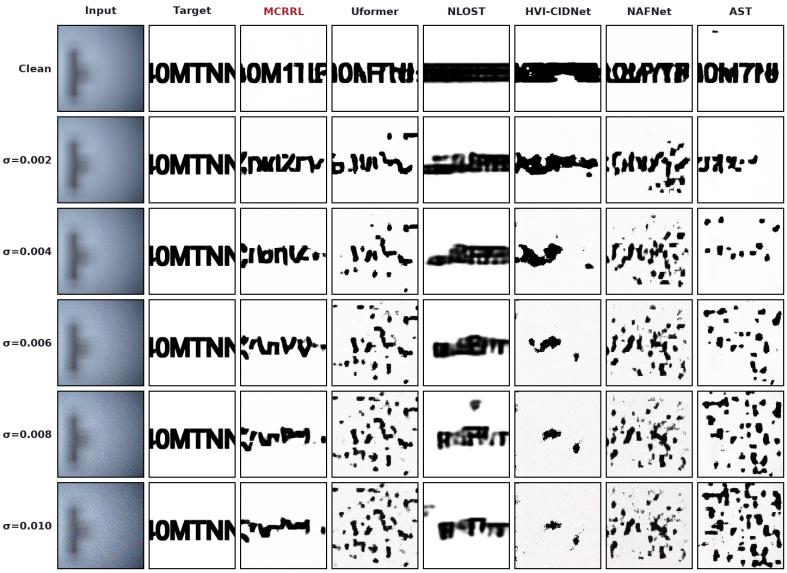

Figure S5: One Password2 example under increasing Gaussian noise, with conditions shown by row and inputs, targets, MCRRL, and baselines shown by column.

Moving downward across the condition rows, the input noise grain becomes visible. Under milder noise, portions of the glyph outlines remain visible and the background is comparatively clean in the proposed output.

Supplementary Figure S6 applies salt-and-pepper noise to Password2 and shows how isolated corrupted pixels alter stroke patterns and black-and-white boundaries.

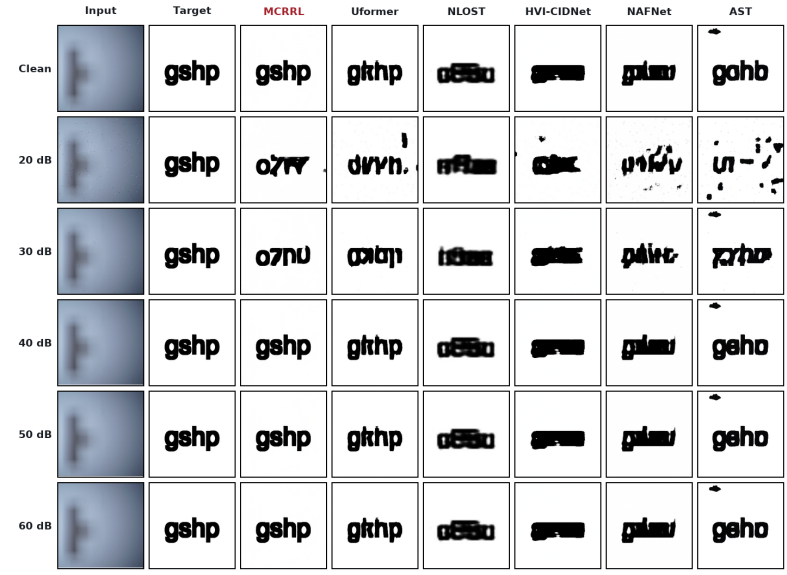

Figure S6: One Password2 example under salt-and-pepper noise. Rows show the clean reference followed by 20, 30, 40, 50, and 60 dB; higher signal-to-noise ratios correspond to milder corruption.

Foreground glyph structure remains recoverable despite isolated corruption in the wall image. In Figure S6, MCRRL preserves the four-character pattern at 40, 50, and 60 dB, with distinct glyph shapes against the white background. Several comparison outputs merge strokes into dark bands or alter their arrangement. This makes the recovery of structured foreground content directly visible under impulse corruption.

Supplementary Figure S7 shows how brightness shifts and additive bias alter visible text strokes and background artifacts.

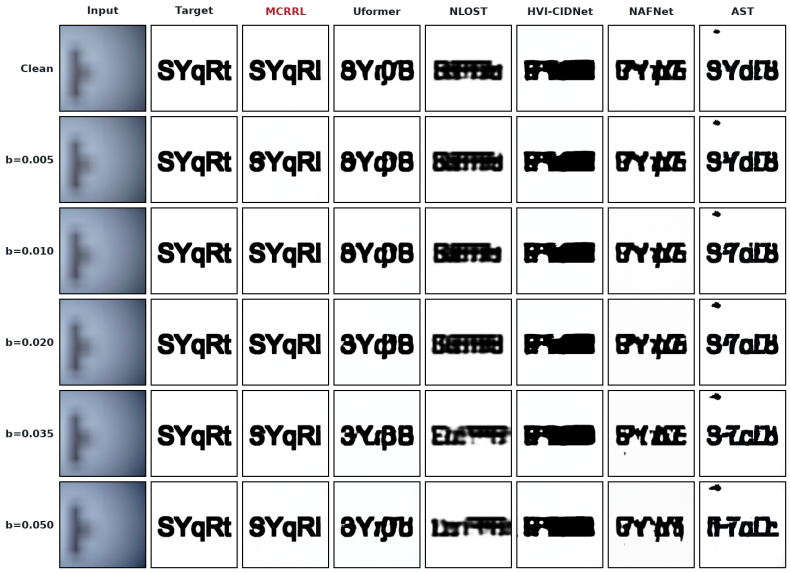

Figure S7: One Password2 example under increasing additive intensity bias.

The brightness reference can shift while the foreground arrangement remains recoverable. Figure S7 shows the main stroke pattern across several additive-bias levels, with foreground and background remaining separated in the milder conditions; the sweep extends to 0.050. On Password2, the normalized SSIM area of 0.91436 and the minimum value of 0.90846 complement the example with dataset-level evidence. Together, the image and summary connect foreground–background recovery to changes in the wall image's intensity reference.

C.3 Relative-Distance Compression Conditions

Screen organization remains recoverable from spatially compressed wall evidence. The relative-distance operator reduces the support available in the observation while leaving the target screen unchanged. Figures S8–S10 examine this change through complete-split curves and representative reconstructions. MCRRL retains major interface partitions, color regions, and prominent wallpaper or window boundaries across multiple compression levels. The combined evidence shows how the reconstruction draws screen structure from an increasingly compact wall-light pattern.

The evaluation holds the other digital operators fixed and reports the complete model's response to spatial-resolution loss.

Supplementary Figure S8 presents primary-metric trajectories across the four datasets as relative-distance compression increases.

Supplementary Figure S9 compares the preservation of interface layouts and local structures across methods as compression severity increases.

The proposed output in this sample retains primary interface partitions and major color blocks at the highest compression level. The visible menu and desktop contours remain coherent as the retained spatial evidence decreases.

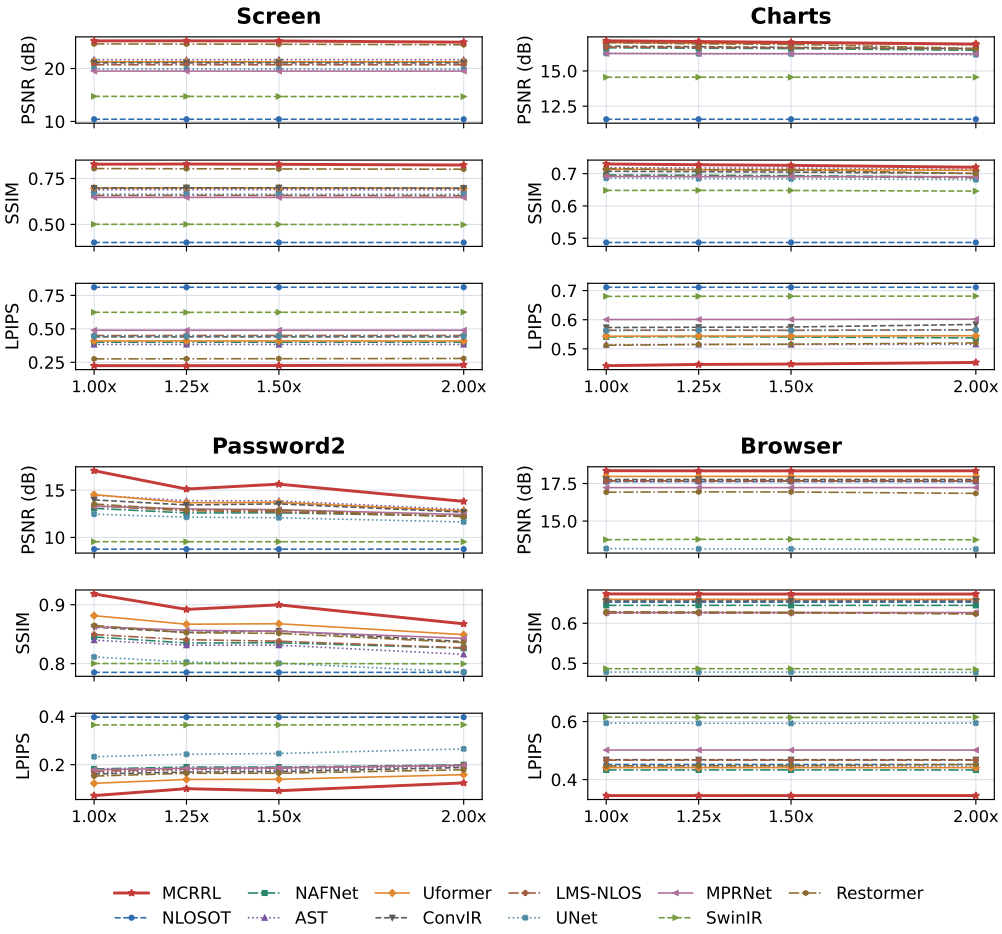

Figure S8: PSNR, SSIM, and LPIPS across relative-distance compression levels on all four datasets.

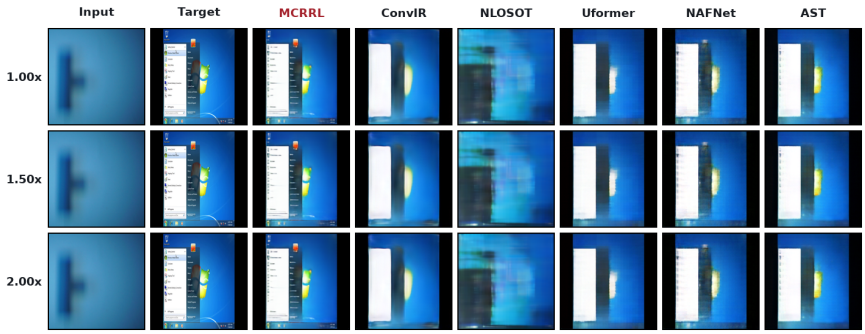

Figure S9: One Screen example under three relative-distance compression levels, comparing MCRRL with external baselines.

Supplementary Figure S10 extends the comparison to stronger relative-distance compression and shows how global layout changes as the input support region shrinks.

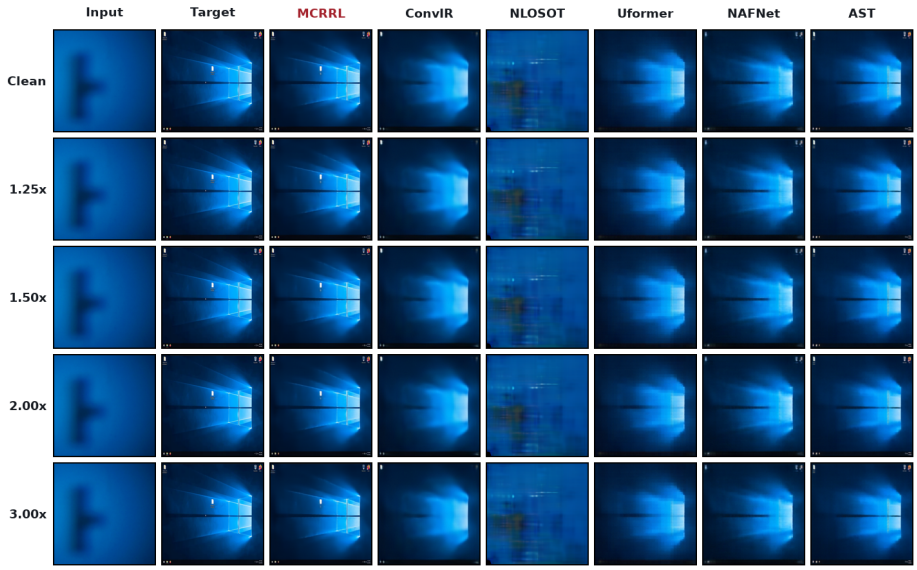

Figure S10: One Screen example over an extended relative-distance compression range.

The extended severity levels show the limits of structural recovery when input evidence is severely compressed. The displayed proposed output retains portions of the main light bands and window contours, while the external outputs show varying regional blur and edge loss.

C.4 Horizontal and Vertical View-Angle Transformations

We examine how perspective changes in the wall-light pattern affect the recovery of screen structure. Figures S11–S12 first consider horizontal transformations, Figures S13–S14 then consider vertical transformations, and Figure S15 combines pitch, yaw, and roll. This sequence moves from changes along one viewing axis to their combined effect.

Supplementary Figure S11 presents primary-metric trajectories across the four datasets as horizontal view-angle transformation severity changes.

The plotted trajectories show the complete-model sensitivity to horizontal perspective transformations. The response varies across the evaluated yaw angles, with structural and perceptual fidelity changing as the image-space perspective transformation becomes more severe.

Supplementary Figure S12 compares the preservation of the main window contours and desktop partitions across methods under horizontal view-angle transformations.

The visual comparison illustrates how horizontal perspective changes affect the reconstruction of digital interfaces. The proposed complete system retains the main window contours and desktop partitions across the displayed conditions.

Supplementary Figure S13 presents primary-metric trajectories across the four datasets as vertical view-angle transformation severity changes.

The plotted trajectories reveal the impact of vertical perspective transformations on complete-model reconstruction quality. Structural and perceptual fidelity vary across the evaluated pitch settings as the image-space perspective transformation becomes more severe.

Supplementary Figure S14 compares the preservation of the main color bands and background partitions across methods under vertical view-angle transformations.

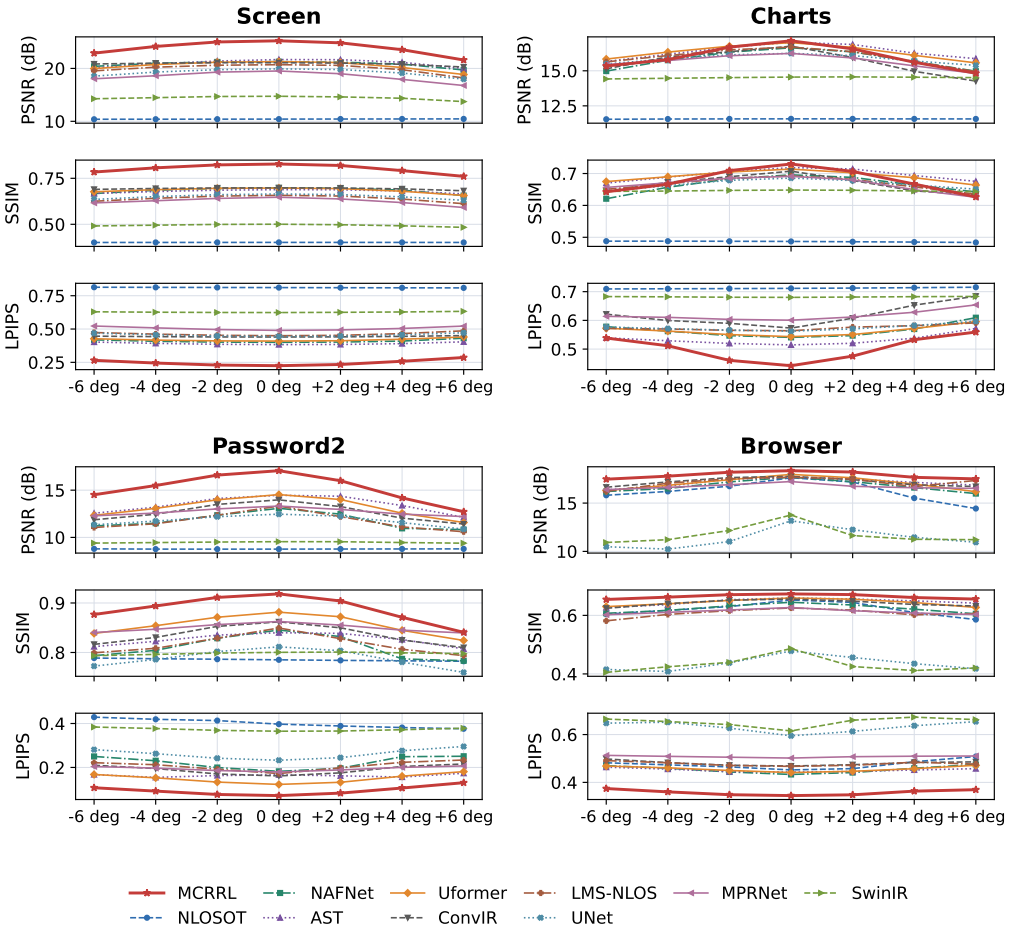

Figure S11: PSNR, SSIM, and LPIPS across horizontal view-angle transformations on all four datasets.

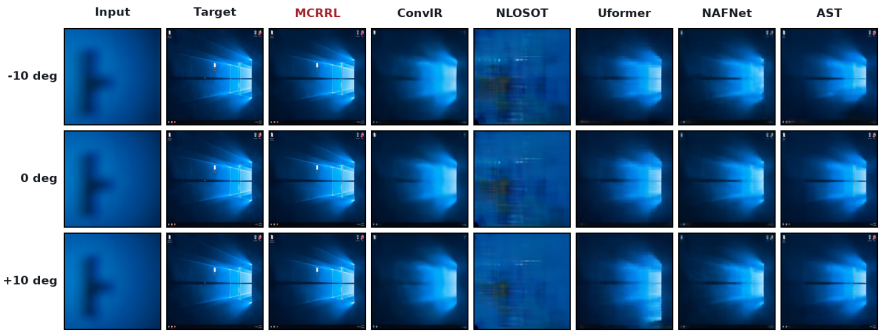

Figure S12: One Screen example under horizontal view-angle transformations, comparing MCRRL with external baselines.

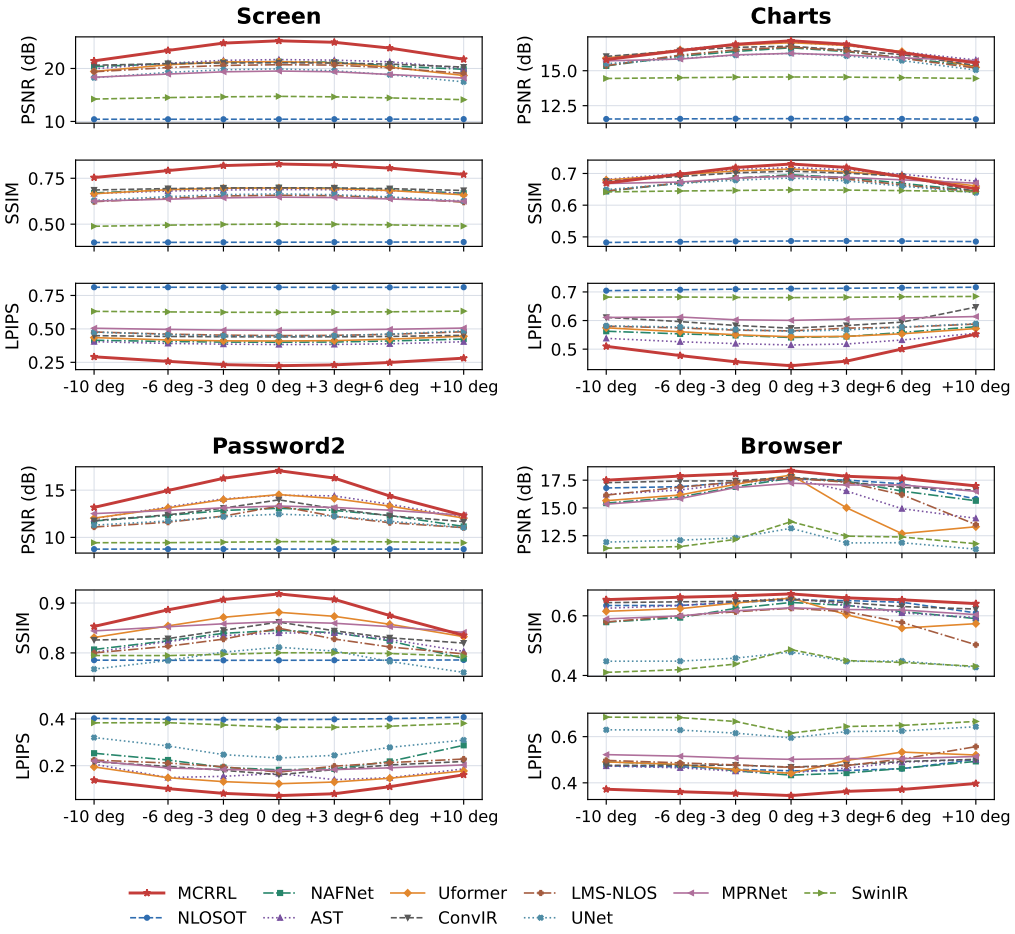

Figure S13: PSNR, SSIM, and LPIPS across vertical view-angle transformations on all four datasets.

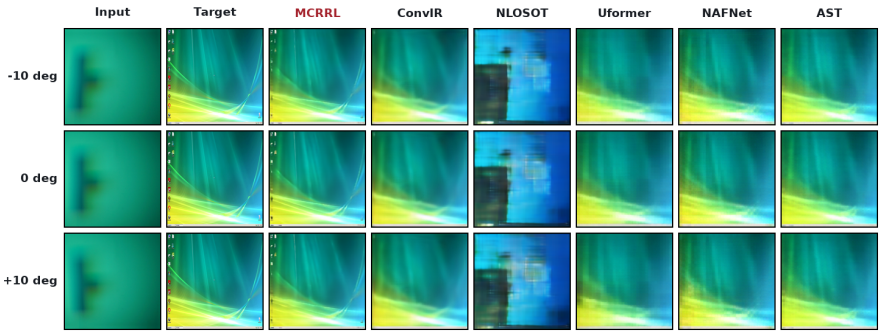

Figure S14: One Screen example under vertical view-angle transformations, comparing MCRRL with external baselines.

MCRRL recovers the principal color bands and their boundaries under the vertical transformations in Figure S14. The wall pattern changes its spatial arrangement, while the output retains the large-scale organization of the target screen.

Supplementary Figure S15 extends the visual comparison to combined pitch, yaw, and roll transformations that alter window positions and regional proportions.

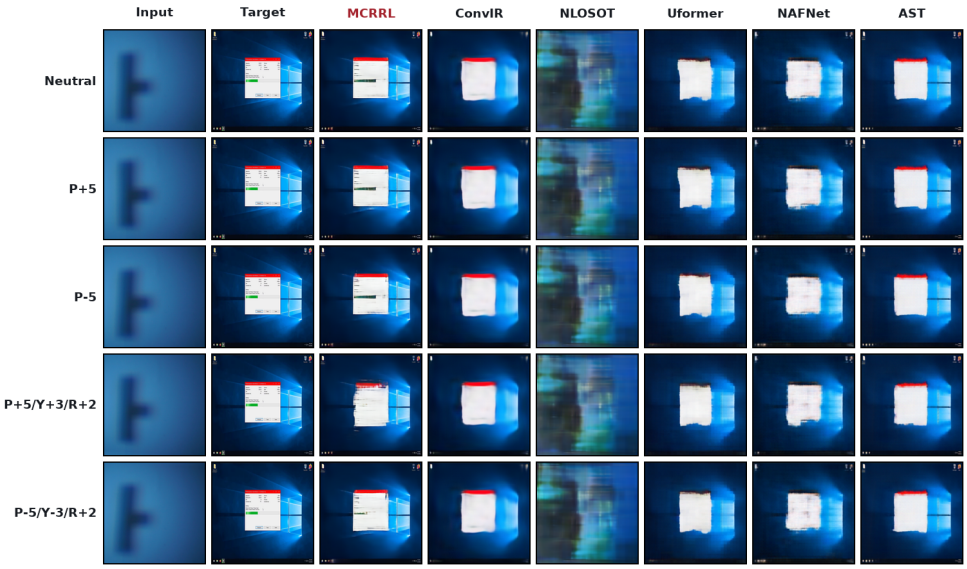

Figure S15: One Screen example under combined pitch, yaw, and roll transformations.

The combined transformations test whether this recovery persists when several viewing axes change together. MCRRL preserves the dialog and desktop layout across the examples in Figure S15, extending the evidence from individual-axis changes to a more compound geometric perturbation.

C.5 Horizontal and Vertical Position Shifts

The position of the wall-light pattern provides a direct test of the learned wall-to-screen correspondence. We translate the observation horizontally and vertically while keeping its screen target fixed. Figures S16–S19 connect these displacements to changes in the recovered desktop layout and local boundaries. The two axes reveal how reconstruction uses the spatial arrangement of the wall evidence.

To visualize the continuous degradation trajectories, Supplementary Figure S16 plots the primary metrics across the entire range of horizontal shift severities.

The curves cover horizontal shifts from -6% to $+6\%$, with the target screen fixed throughout.

The qualitative impact of these horizontal shifts on the reconstructed digital screen structures is detailed in Supplementary Figure S17.

At the milder horizontal shifts, MCRRL preserves the desktop color layout, foreground dialog, and its principal boundaries. The recovered regions maintain their screen-space arrangement despite the displacement of the wall observation.

The corresponding continuous degradation trajectories for the vertical dimension are plotted in Supplementary Figure S18.

Supplementary Figure S18 shows a vertical-shift profile that differs from the horizontal profile. PSNR declines and LPIPS rises as the displayed vertical shift moves away from zero, while the gap to the external curves varies by condition.

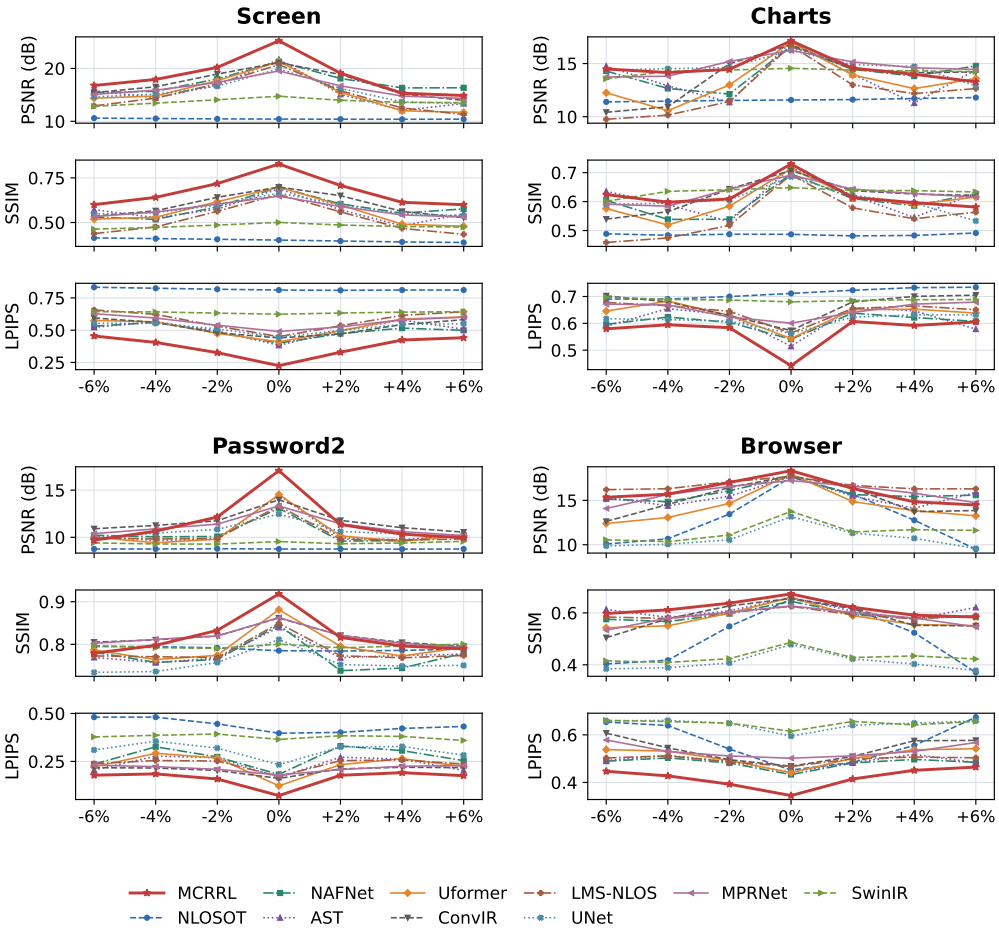

Figure S16: PSNR, SSIM, and LPIPS across horizontal-position shifts on all four datasets.

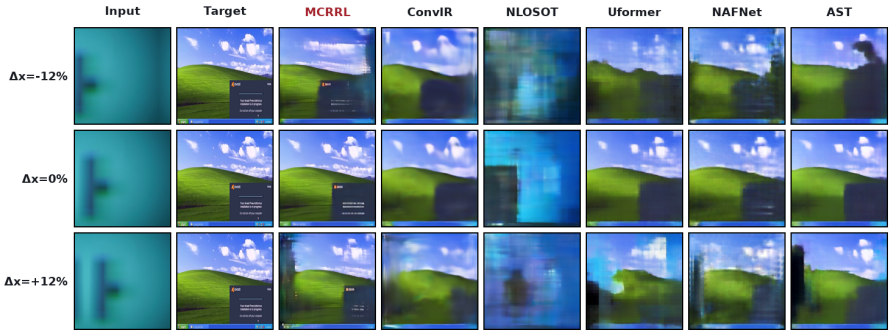

Figure S17: Representative Screen outputs under horizontal-position shifts; Charts is covered by the companion metric curves.

Finally, Supplementary Figure S19 provides the qualitative visual panels for the vertical-shift evaluation.

The vertical-shift examples retain the main desktop regions and the separation between the foreground dialog and its background. Together with the horizontal results, they connect position changes in the wall measurement to the reconstruction of coherent screen layout.

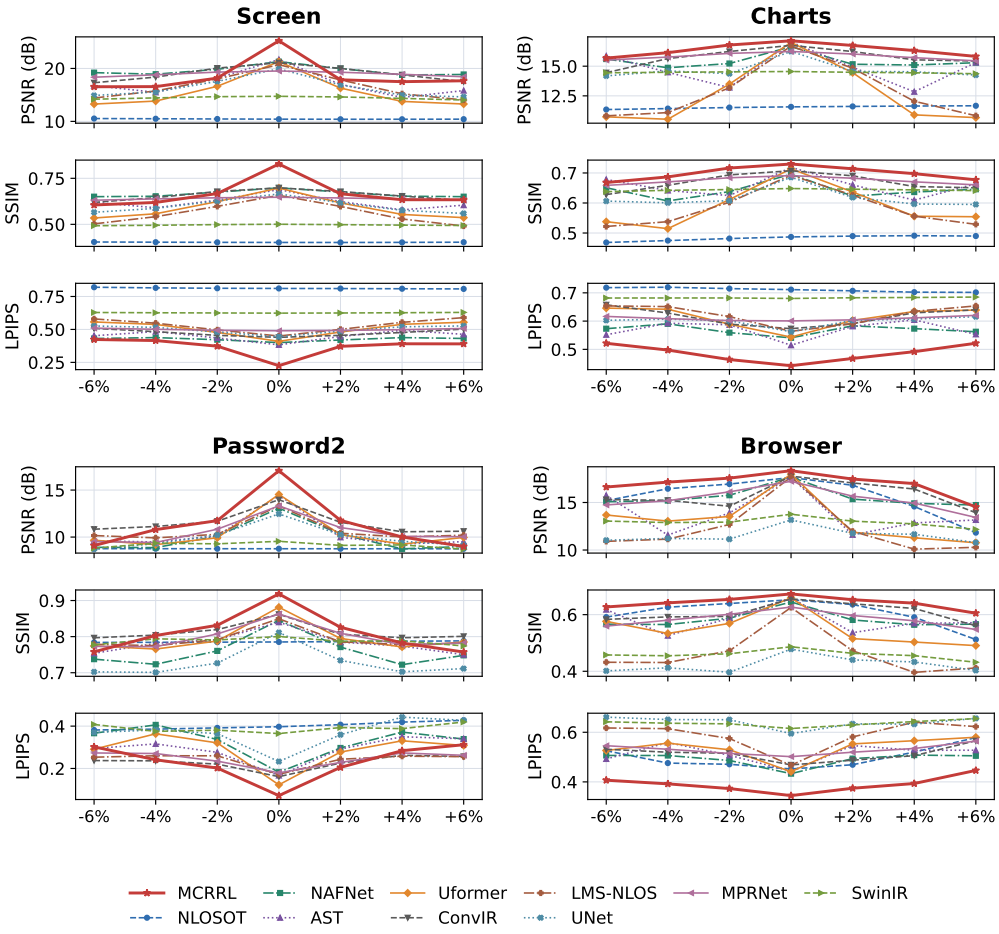

Figure S18: PSNR, SSIM, and LPIPS across vertical-position shifts on all four datasets.



Figure S19: Representative Screen outputs under vertical-position shifts; Charts is covered by the companion metric curves.

C.6 Controlled Wall-Response and Reflectance Variations

Beyond geometric misalignments, hidden-screen reconstruction depends on the photometric properties of the reflecting surface. Diffuse reflection attenuates spatial detail and changes the contrast, color saturation, and signal-to-noise ratio of the indirect observation. Supplementary Figures S20–S24 show the curves and recovery panels for controlled wall-response and reflectance operators across the four paired datasets.

Supplementary Figures S20–S21 show the complete-model response across the wall-response conditions spanning the four datasets.

In addition to the categorical wall-response operator, we evaluate a continuous reflectance-response operator that scales the overall intensity of the indirect observation. Figures S22–S23 show the reflectance-response curves and recovery panels.

To visualize the impact of the categorical wall-response operator across the four datasets, Supplementary Figure S20 plots the primary metrics for each wall-response condition.

Supplementary Figure S20 reports the complete-system response across four wall-response operators. The proposed SSIM curve is above the displayed external curves for the four datasets, while the largest displayed drop occurs under the plaster operator.

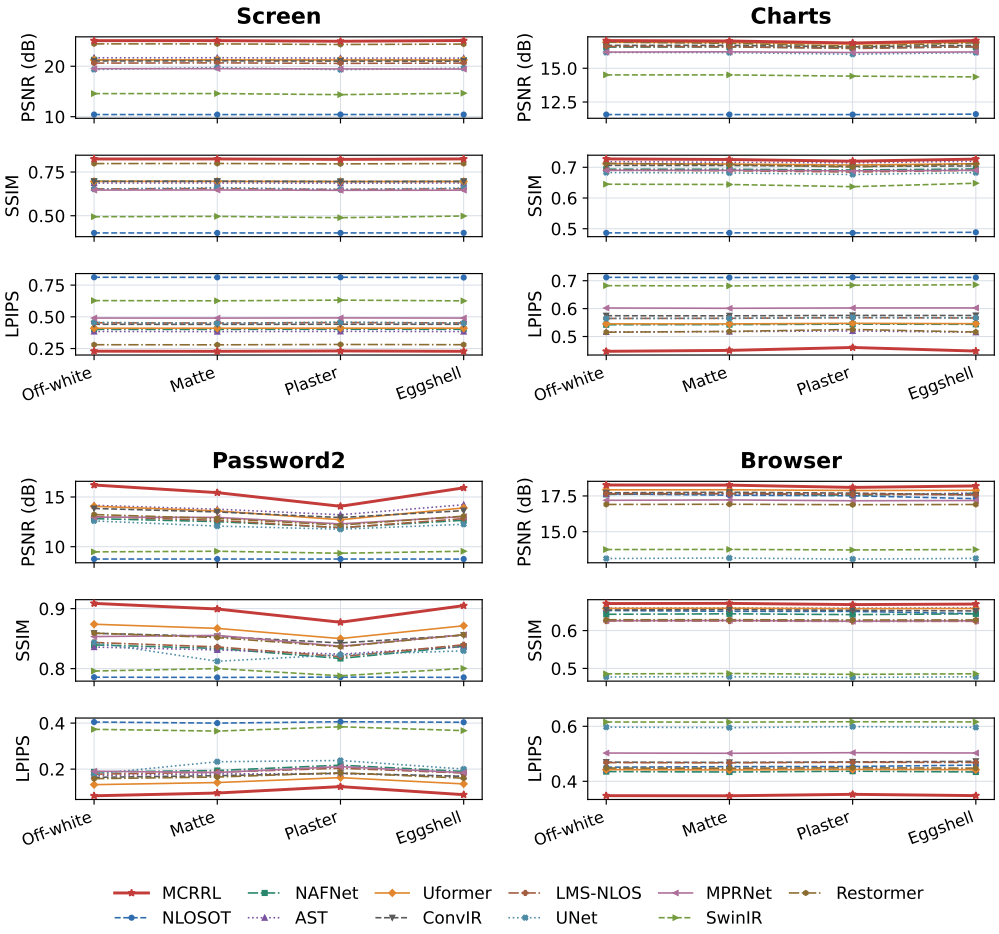

Figure S20: PSNR, SSIM, and LPIPS across four wall-response operators on all four datasets.

The qualitative effects of the wall-response operator on a specific natural image layout are detailed in Supplementary Figure S21.

Changes in wall response can alter the appearance of the measurement while leaving the screen's organization recoverable. In Figure S21, MCRRL preserves the sky–ground division, curved hill contour, and bottom taskbar across the wall-response conditions. These relationships connect the output to the target's spatial layout even as the observation changes in color and contrast.

The continuous degradation trajectories for the reflectance-response operator are plotted in Supplementary Figure S22.

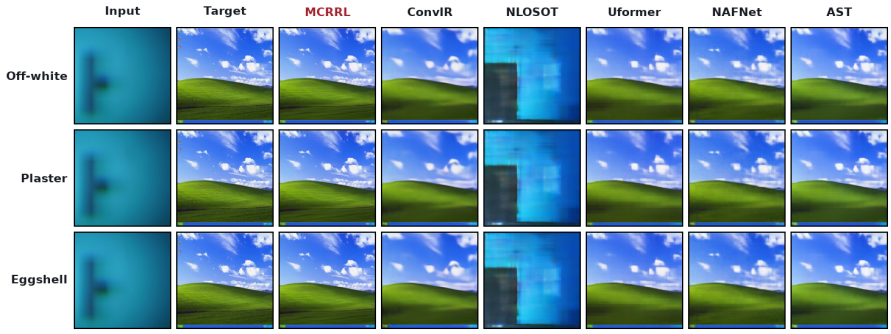

Figure S21: One Screen example under three wall-response operators.

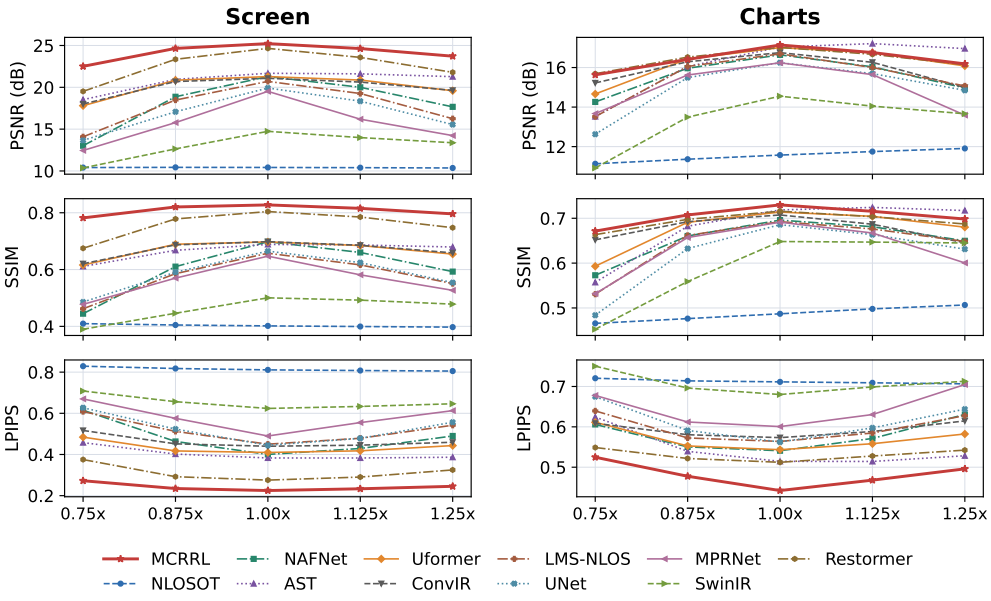

Figure S22: PSNR, SSIM, and LPIPS across reflectance-response scale factors on Screen and Charts.

Supplementary Figure S22 reports a roughly symmetric complete-system metric decline around the unperturbed factor of 1.00. The displayed gap to external curves changes only modestly across the mild range, while PSNR drops more at the extreme factors.

The qualitative visual panels for the reflectance-response evaluation are provided in Supplementary Figure S23.

The reflectance-response sweep separates changes in measurement intensity from changes in spatial arrangement. The operator rescales the wall observation while preserving its underlying layout, allowing us to examine how screen reconstruction responds to a changed brightness reference. Across the qualitative conditions in Figure S23, MCRRL retains the major color partition and foreground dialog. Figure S22 quantifies the accompanying fidelity changes, connecting photometric variation to the recovery of screen structure.

Finally, Supplementary Figure S24 provides an additional qualitative analysis of the wall-response operator on a different visual layout.

The visual panels in Supplementary Figure S24 feature a Vista desktop with a prominent green light band and a cyan background, providing a continuous large-area color and curved-boundary reference. This experiment reports how the stated changes in reflectance, large-area hue, and surface texture affect color transfer and structural recovery.

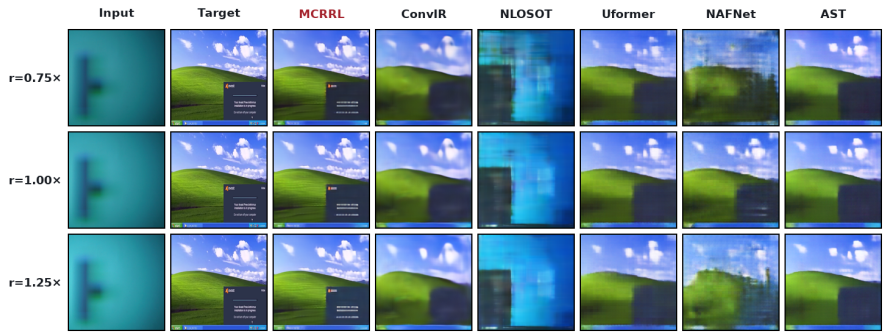

Figure S23: Representative Screen outputs under three reflectance-response conditions; Charts is covered by the companion metric curves.

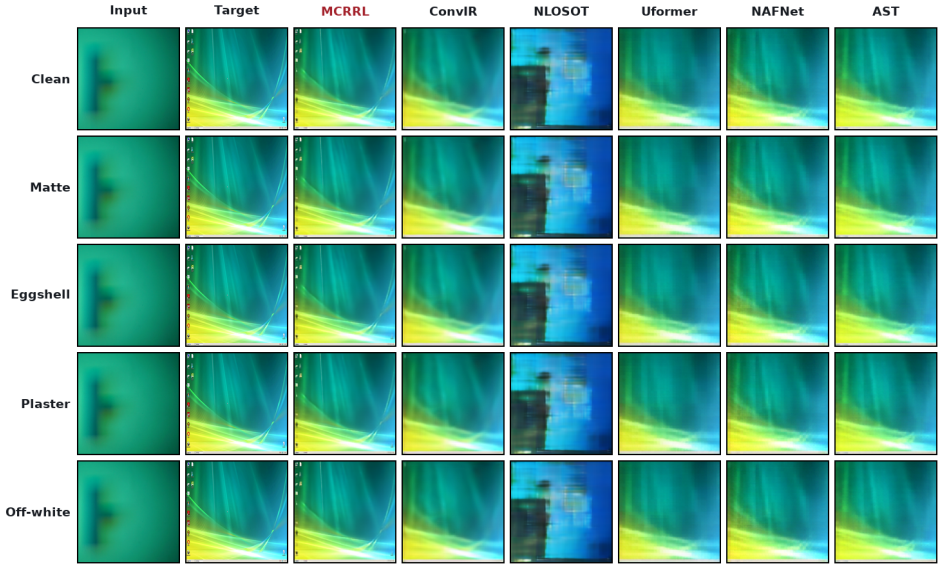

Figure S24: One Screen example under clean and four wall-response conditions.

C.7 Controlled Ambient-Light Direction and Strength Variations

This section evaluates the stability of the proposed method under controlled ambient-light perturbations. The evaluation applies deterministic image-space operators that vary the direction and strength of the ambient-light field.

Supplementary Figures S25–S27 show the primary curves and recovery panels for ambient-direction and ambient-strength conditions. The curves show limited metric variation under mild ambient-light conditions and greater degradation at higher severities.

Supplementary Figure S25 plots the primary metrics across ambient-direction conditions.

Supplementary Figure S26 reports the continuous response to ambient-light strength.

The strength curves reveal a gradual decline in reconstruction performance as the severity of the ambient-light operator increases. The proposed complete system maintains a measurable advantage over the external baselines in the low to moderate strength regimes.

Supplementary Figure S27 compares how the methods handle spatial color shifts introduced by ambient-light conditions on one Screen sample.

Ambient illumination changes the brightness and color field through which the screen's weak wall signature is observed. Across the six conditions in Figure S27, MCRRL retains the large blue desktop region, bright window-shaped wallpaper motif, and bottom taskbar. The

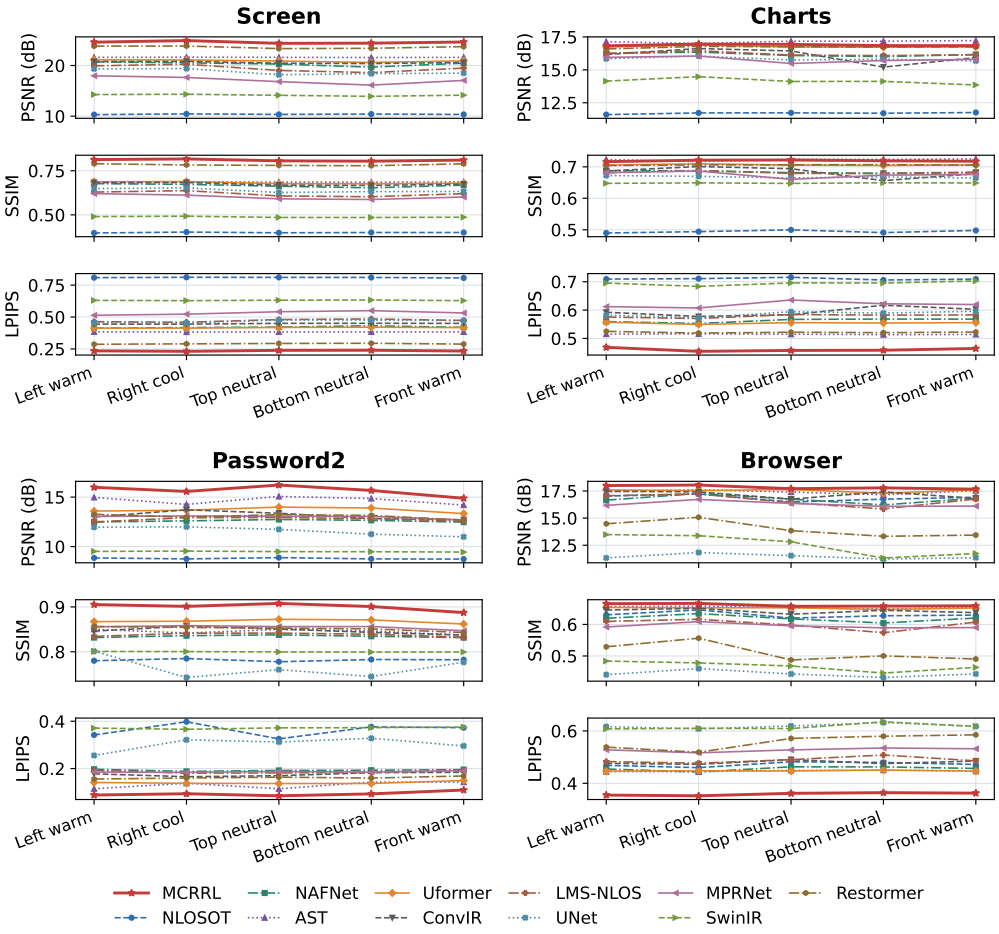

Figure S25: Metric curves across ambient-direction conditions on all four datasets.

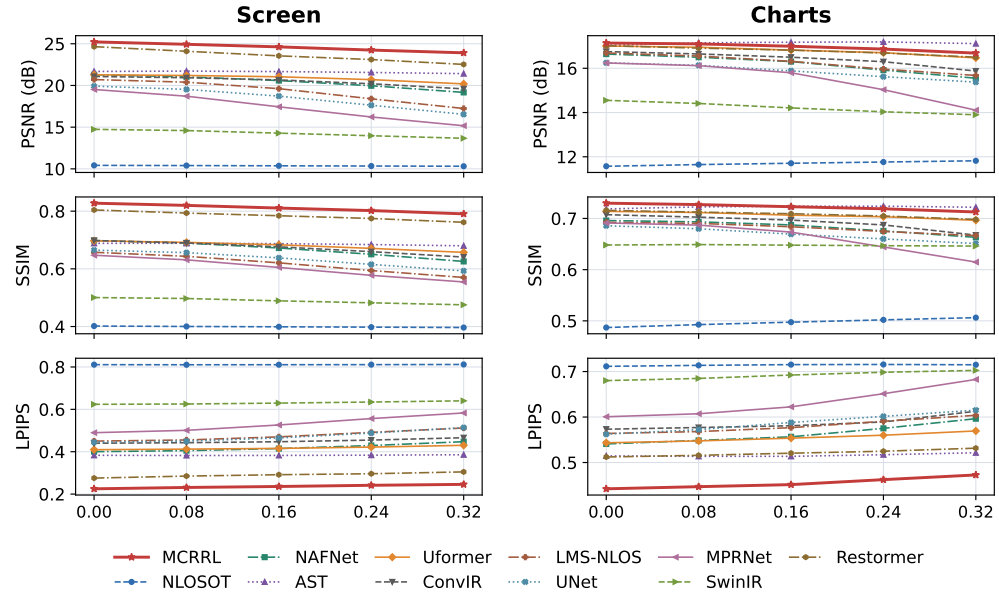

Figure S26: Metric curves across ambient-strength conditions on Screen and Charts.

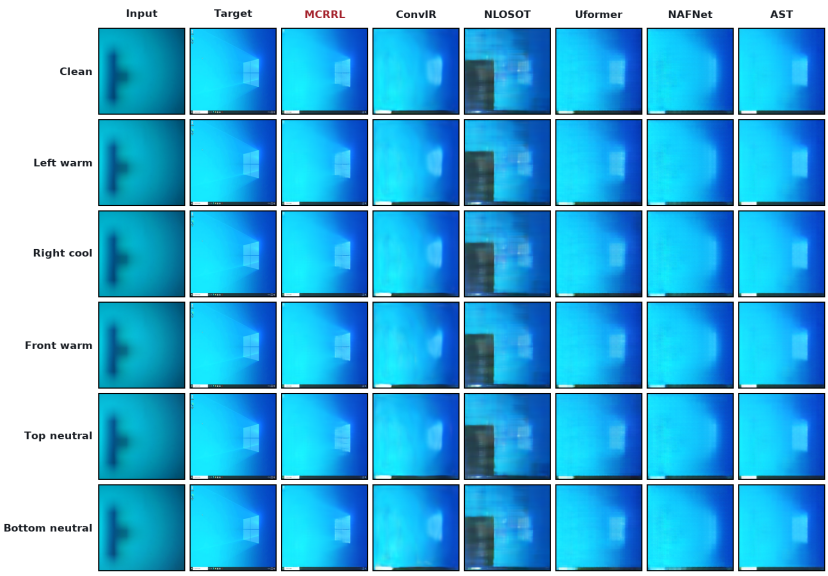

Figure S27: One Screen example under six ambient-light conditions, comparing MCRRL with external baselines.

direction and strength sweeps complement these examples by following the reconstruction scores as the added illumination changes. The results connect resistance to photometric variation with the recovery of a consistent screen layout.

C.8 Controlled Lens-Effect Operators

Controlled lens perturbations test reconstruction stability when optical distortions alter the wall evidence. The evaluation applies deterministic image-space operators representing geometric distortion, chromatic aberration, and defocus blur.

Supplementary Figures S28–S30 show the complete-system response under the reported lens perturbations, with performance varying by perturbation type. The evaluation fixes the model weights, target images, test splits, and evaluation procedures, changing only one input condition at a time.

Supplementary Figure S28 plots the primary metrics across lens-effect conditions and datasets.

The curves show the complete-system metric variation across the lens-effect conditions, particularly on the Screen and Password2 datasets.

Supplementary Figure S29 compares method outputs under defocus and chromatic-aberration operators on one Screen sample.

The layered window arrangement remains identifiable under the defocus and chromatic-aberration operators. MCRRL preserves the blue selection bar and major panel boundaries, connecting the reconstruction to the target interface despite changes in the wall image's sharpness and channel alignment.

Supplementary Figure S30 extends the comparison to geometric distortion on a second Screen sample.

The second example extends the test to geometric distortion. The reconstructed blue desktop, start-menu region, and bright wallpaper motif retain the main screen organization across the lens-effect conditions.

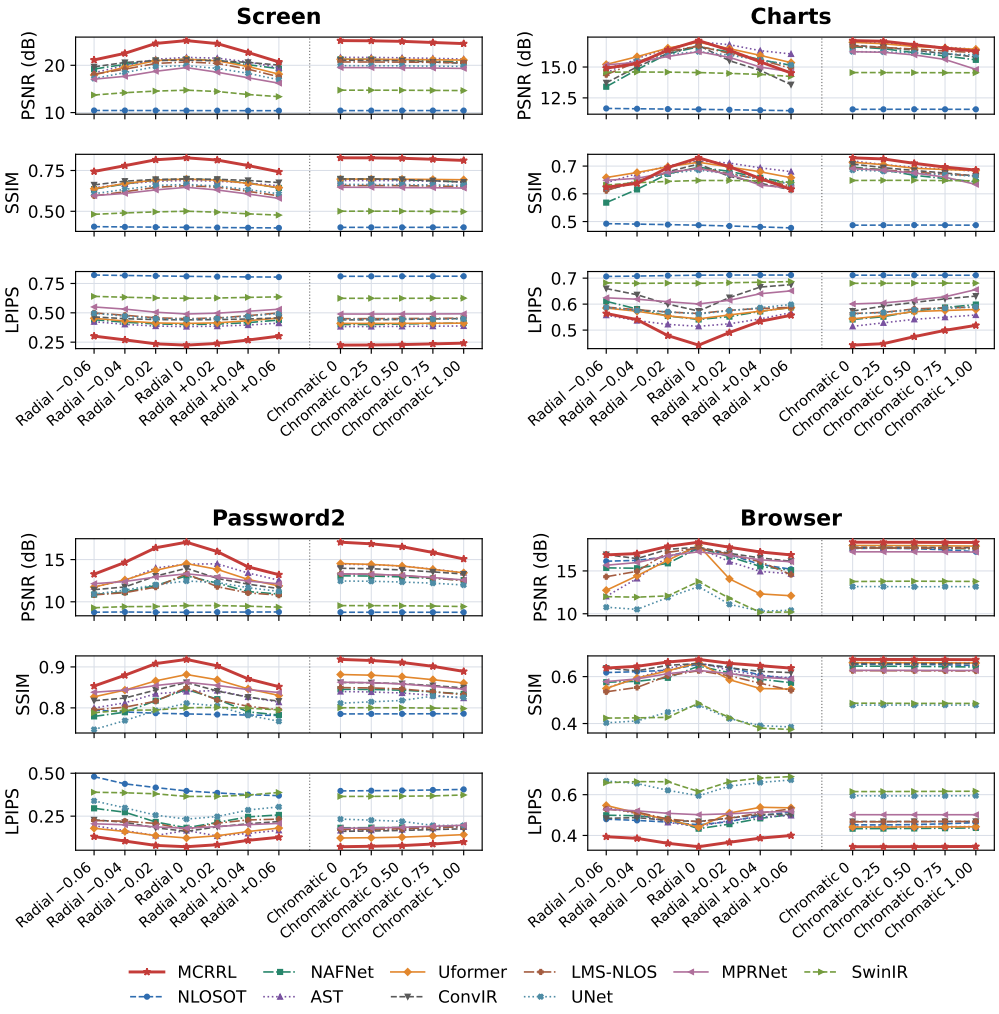

Figure S28: Metric curves across lens-effect operators on all four datasets.

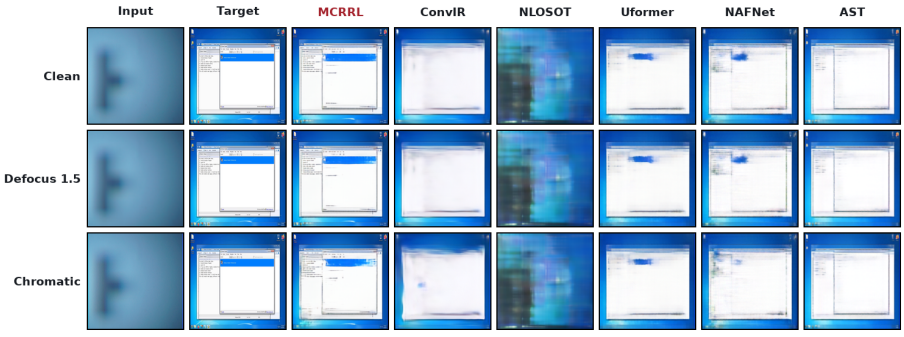

Figure S29: One Screen example under clean, defocus, and chromatic-aberration conditions.



Figure S30: One Screen example under geometric-distortion, chromatic-aberration, and defocus conditions.

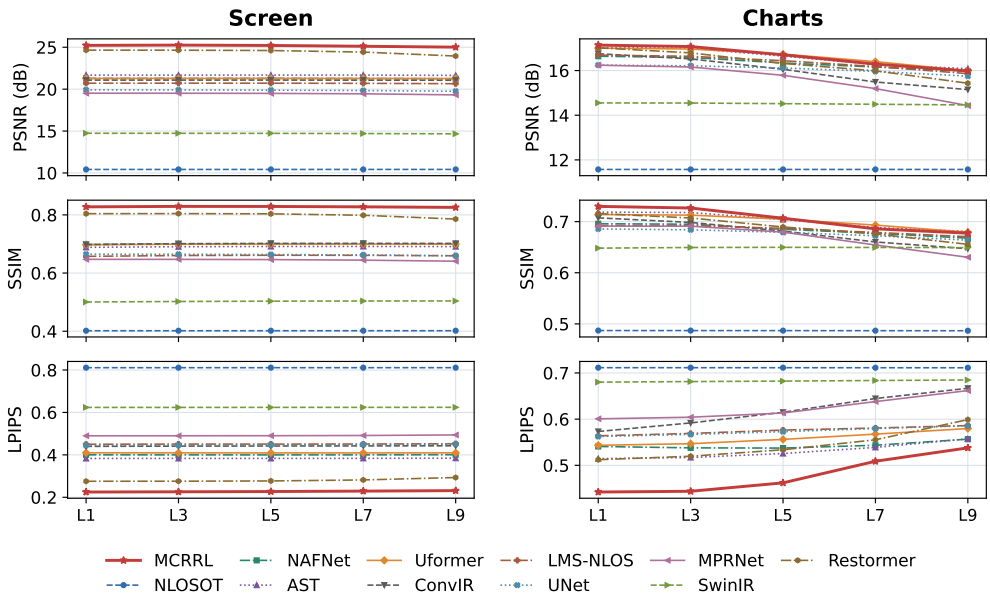

Figure S31: PSNR, SSIM, and LPIPS across motion-blur kernels on Screen and Charts.

C.9 Controlled Motion Blur

We vary motion-kernel length and direction to examine how spatial spreading of the wall-light pattern affects screen reconstruction. Motion blur mixes the wall evidence along a chosen direction, linking the kernel directly to the spatial cues available for reconstruction. Figures S31–S33 follow this change from metric curves to desktop partitions, window boundaries, and text-line structure. Under the shorter kernels, MCRRL preserves the principal interface arrangement and visible foreground organization. The length and direction sweeps then show how that recovery responds as the wall-light pattern spreads further.

The visual panels connect motion-induced spatial spreading to the recovery of supported screen structure under the reported image-space operators. In these displayed samples, the



Figure S32: Representative Screen outputs under three motion-blur conditions; Charts is covered by the companion metric curves.

proposed outputs contain more visible structural detail than the compared baseline outputs, which appear more strongly smoothed under the motion-blur conditions.

Supplementary Figure S33 compares horizontal, vertical, and diagonal blur on windows, text blocks, and interface boundaries.

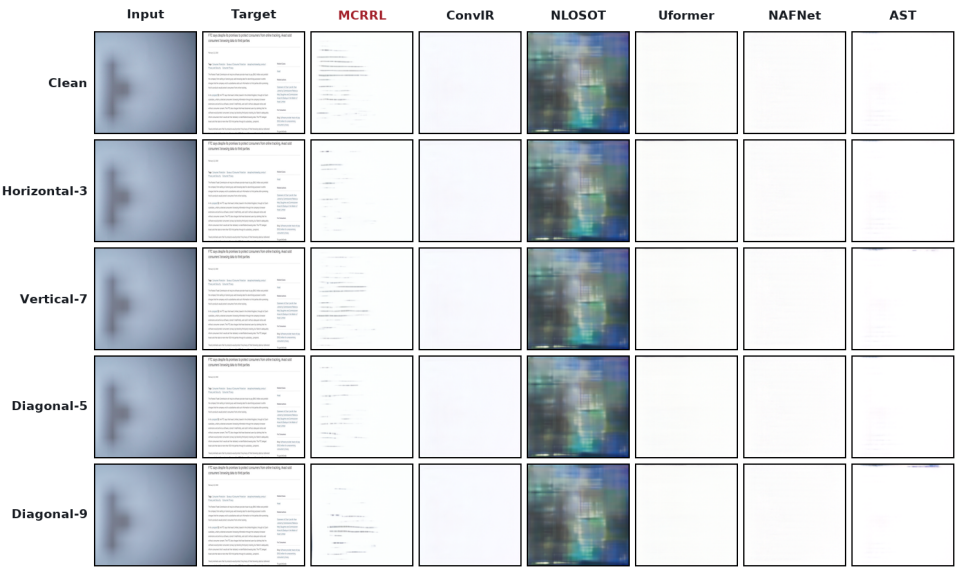

Figure S33: One Screen example under directional blur conditions, comparing MCRRL with external baselines.

This figure compares the effects of motion-kernel direction and length on local lines and global layout. Portions of the interface boundaries remain visible in the proposed output under the shorter kernels.

C.10 Geometry and Material Retention Summary

Supplementary Figures S8, S11, S13, and S20 aggregate the complete-model curves under the same four geometry and material perturbation families for Screen, Charts, Password2, and Browser. Each dataset covers relative distance, horizontal view angle, vertical view angle, and wall response.

Geometry and wall response change different aspects of the evidence used to recover the screen. Geometric transformations alter where the wall cues appear and how they are spatially compressed; wall-response operators alter their brightness, color, and contrast. The recovered examples connect both kinds of change to the preservation of screen layout and foreground organization. Taken together, Figures S8, S11, S13, and S20 relate reconstruction

quality to the structure and appearance of the wall measurement. The following learning and spatial diagnostics examine how MCRRL uses these cues to form the final screen image.

C.11 Fixed-Representation Learning Comparison Under Stratified Conditions

Both subsets use the paired Screen test split after bicubic resizing to 256×256. The high-detail subset contains samples at or above the 75th percentile of $\langle|\partial_h \bar{y}|\rangle + \langle|\partial_v \bar{y}|\rangle$, where $\bar{y}$ is target grayscale formed by averaging RGB and the differences are averaged over valid adjacent pixel pairs; the large-gap subset uses the same percentile rule on $\langle|x - y|\rangle$.

Supplementary Table S9 compares learning objectives on the high-detail target quartile.

Table S9: Learning-objective comparison on the high-detail Screen subset defined above, under a fixed network interface. PSNR and SSIM are higher is better; MSE, RMSE, LPIPS, and Edge-MAE are lower is better. Red and blue values mark the best and second-best result within each metric; row fills identify rows containing at least one marked value.

Objective	PSNR ↑	MSE ↓	RMSE ↓	SSIM ↑	LPIPS ↓	Edge-MAE ↓
Pixel L1	13.4019	0.0550963	0.224277	0.414291	0.721477	0.0717996
Charbonnier	13.6953	0.0519369	0.217113	0.422997	0.708004	0.0718194
L1 + MS-SSIM	13.729	0.0534132	0.218269	0.426912	0.712286	0.0710609
L1 + LPIPS	13.9064	0.0519853	0.214436	0.401754	0.603378	0.0759328
Composite (Ours)	13.9253	0.0516752	0.21394	0.438347	0.667192	0.070422

Fine screen structure benefits from coupling pixel agreement with structural and boundary constraints. On the high-detail subset, the composite objective achieves the highest PSNR and SSIM together with the lowest MSE, RMSE, and Edge-MAE. These joint gains connect color fidelity to the preservation of detailed spatial organization.

Supplementary Table S10 compares learning objectives on the quartile with the largest observation-target gap.

Table S10: Learning-objective comparison on the large observation–target-gap Screen subset defined above, under a fixed network interface. PSNR and SSIM are higher is better; MSE, RMSE, LPIPS, and Edge-MAE are lower is better. Red and blue values mark the best and second-best result within each metric; row fills identify rows containing at least one marked value.

Objective	PSNR ↑	MSE ↓	RMSE ↓	SSIM ↑	LPIPS ↓	Edge-MAE ↓
Pixel L1	15.4619	0.042587	0.187159	0.564352	0.557007	0.0464241
Charbonnier	15.7901	0.0405869	0.181263	0.575753	0.542844	0.0461961
L1 + MS-SSIM	16.778	0.0371033	0.168222	0.590779	0.528692	0.0446378
L1 + LPIPS	16.8045	0.0366009	0.167047	0.572336	0.457275	0.0489623
Composite (Ours)	17.0293	0.03679	0.166046	0.618577	0.493168	0.0437124

The large observation–target-gap subset also shows gains in PSNR, SSIM, RMSE, and Edge-MAE. The composite objective reaches 17.0293 dB PSNR, 0.618577 SSIM, and 0.0437124 Edge-MAE, with the lowest RMSE of 0.166046. This agreement across reconstruction and structure measures supports combining complementary training signals when the wall image differs strongly from the hidden screen.

MPRF shows a perceptual-stability advantage when part of the wall evidence is removed. In the fixed-U-Net comparison, it produces the smallest relative LPIPS increase at the nonzero partial-support-loss levels. This result connects the structured representation to reconstruction from the remaining observation support. The weak-measurement and scale-compression panels describe the responses to two other forms of evidence change.

The curves use every reported sample in the complete paired Screen test split. The three columns sequentially display the weak measurement, the partial support loss, and the scale

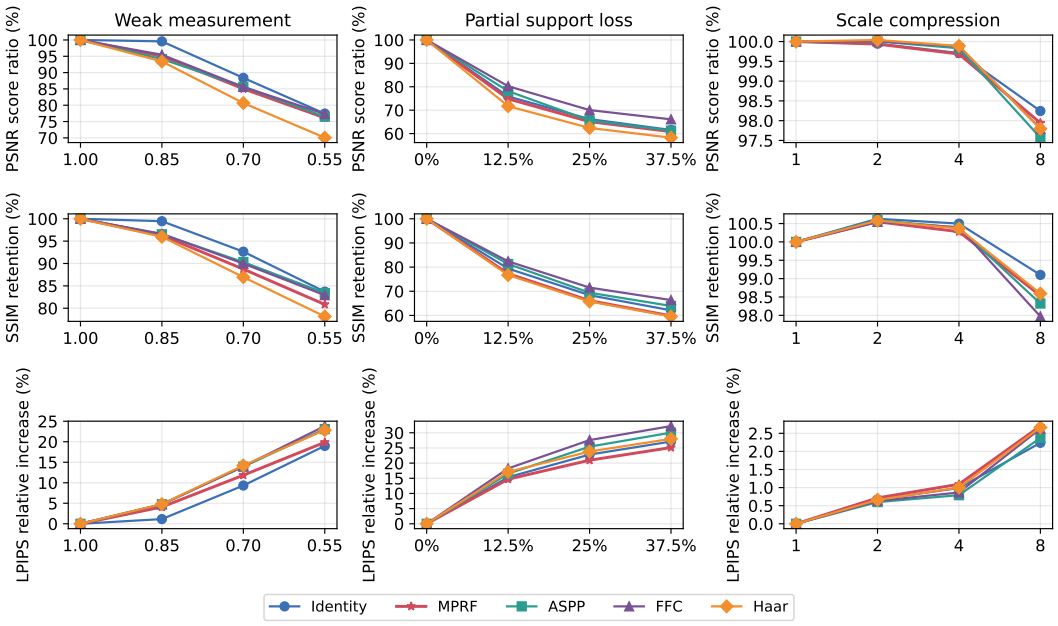

Figure S34: Retention and degradation curves for five representations in a fixed U-Net under three evidence-loss families on Screen. The variants compare representation mechanisms, including MPRF, rather than complete MCRRL models.

compression conditions, while the three rows sequentially display the PSNR score ratio, the SSIM retention, and the relative LPIPS increase.

C.12 Nine-Grid Spatial Sensitivity

Screen reconstruction depends unevenly on different regions of the wall observation. Tables S11–S14 report the complete paired-test nine-grid sensitivity results for Screen, Charts, Password2, and Browser, respectively. The 216 intervention scores examine this dependence from two directions: removing a region measures how much the reconstruction relies on it, while retaining a region measures the structures that can be recovered from that part alone. On Screen, occlusion SSIM ranges from 0.41703 to 0.79308, and single-region retention reaches 0.42297. The contrast between the two interventions supports combining information across wall regions to reconstruct the screen's overall organization.

Table S11: Nine-grid spatial sensitivity on the complete paired Screen test split. Occlusion and retention blocks follow the 3×3 grid; Clean is an unranked reference under the same settings. Within each block, red/blue text marks the best/second-best value, and pale-red/pale-blue bands mark the best/second-best grid-row mean.

Metric	Grid row	Occlusion			Single-region retention			Clean
		Left	Center	Right	Left	Center	Right	
SSIM ↑	Top	0.67125	0.59006	0.77208	0.38263	0.3093	0.42297	0.82767
	Middle	0.4217	0.41703	0.73661	0.33968	0.32255	0.38584	
	Bottom	0.67079	0.67553	0.79308	0.35852	0.36798	0.4139	
LPIPS ↓	Top	0.37052	0.44272	0.27905	0.71089	0.71334	0.70205	0.22485
	Middle	0.60569	0.58372	0.31456	0.70439	0.70541	0.71342	
	Bottom	0.38011	0.37103	0.25955	0.70496	0.71084	0.72357	
Edge-MAE ↓	Top	0.042625	0.046596	0.035415	0.052669	0.06159	0.048169	0.031116
	Middle	0.058155	0.056731	0.038541	0.055991	0.060727	0.054612	
	Bottom	0.041915	0.041678	0.033576	0.055532	0.054482	0.050277	

Table S12: Nine-grid spatial sensitivity on the complete paired Charts test split; grid layout, Clean reference, and highlighting conventions follow Supplementary Table S11.

Metric	Grid row	Occlusion			Single-region retention			Clean
		Left	Center	Right	Left	Center	Right	
SSIM ↑	Top	0.41123	0.56599	0.68596	0.47892	0.39558	0.43799	0.72982
	Middle	0.40317	0.47107	0.68278	0.45032	0.36376	0.40216	
	Bottom	0.43647	0.53158	0.69946	0.39376	0.37389	0.45622	
LPIPS ↓	Top	0.68803	0.60058	0.49674	0.70574	0.72118	0.71744	0.44219
	Middle	0.70443	0.67343	0.49938	0.6913	0.71173	0.71996	
	Bottom	0.65945	0.63661	0.48383	0.7282	0.70183	0.73443	
Edge-MAE ↓	Top	0.076133	0.064123	0.051626	0.065005	0.075264	0.072221	0.047758
	Middle	0.074738	0.068962	0.051861	0.072648	0.085142	0.081904	
	Bottom	0.072239	0.063276	0.050265	0.077673	0.07746	0.069516	

Table S13: Nine-grid spatial sensitivity on the complete paired Password2 test split; grid layout, Clean reference, and highlighting conventions follow Supplementary Table S11.

Metric	Grid row	Occlusion			Single-region retention			Clean
		Left	Center	Right	Left	Center	Right	
SSIM ↑	Top	0.85913	0.80966	0.91477	0.74589	0.75711	0.78442	0.91832
	Middle	0.76128	0.79085	0.89844	0.72484	0.73321	0.76961	
	Bottom	0.78114	0.85437	0.88292	0.74424	0.73098	0.76702	
LPIPS ↓	Top	0.14992	0.31713	0.0781	0.39977	0.41828	0.45118	0.07186
	Middle	0.31431	0.29643	0.12016	0.37717	0.51903	0.35922	
	Bottom	0.26253	0.1969	0.11694	0.33389	0.36065	0.36451	
Edge-MAE ↓	Top	0.037047	0.032305	0.027789	0.036638	0.033737	0.026701	0.027439
	Middle	0.042329	0.039398	0.027353	0.042554	0.029121	0.032262	
	Bottom	0.044002	0.032257	0.03098	0.041288	0.042417	0.033684	

Table S14: Nine-grid spatial sensitivity on the complete paired Browser test split; grid layout, Clean reference, and highlighting conventions follow Supplementary Table S11.

Metric	Grid row	Occlusion			Single-region retention			Clean
		Left	Center	Right	Left	Center	Right	
SSIM ↑	Top	0.60014	0.57299	0.66257	0.32819	0.45789	0.27171	0.6733
	Middle	0.56793	0.5506	0.58947	0.32223	0.38367	0.30566	
	Bottom	0.65526	0.58123	0.66365	0.30086	0.3678	0.30604	
LPIPS ↓	Top	0.43457	0.46483	0.36164	0.69652	0.60578	0.7209	0.34474
	Middle	0.46484	0.48292	0.45154	0.68483	0.63779	0.65228	
	Bottom	0.36713	0.47247	0.35914	0.70723	0.64578	0.68722	
Edge-MAE ↓	Top	0.046051	0.047585	0.042251	0.055723	0.050892	0.051814	0.041664
	Middle	0.047469	0.04864	0.046347	0.056254	0.055271	0.056665	
	Bottom	0.042385	0.045937	0.042045	0.054054	0.052469	0.057005	

Supplementary Figure S35 visualizes how occluding different grid regions affects recovery of the blue desktop, the central bright window, and the horizontal light band.

Figure S35 shows how removing different wall regions changes the reconstructed blue desktop, bright wallpaper bands, and small desktop elements. The responses differ across intervention locations, making the spatial dependence of the reconstruction visible.

Supplementary Figure S36 complements the occlusion intervention by showing the partial structures recovered when only one grid region is retained.

Different retained wall regions support different combinations of screen color and layout. Their outputs recover partial structures at multiple screen locations, illustrating how spatially distributed wall evidence contributes to the hidden display reconstruction.

Supplementary Figure S37 aggregates complete-test-set spatial sensitivity and contrasts the numerical effects of occlusion and single-region retention.

Removing the middle-center wall region gives the lowest occlusion SSIM, 0.41703. Retaining the top-right region gives the highest SSIM among the single-region inputs, 0.42297. The

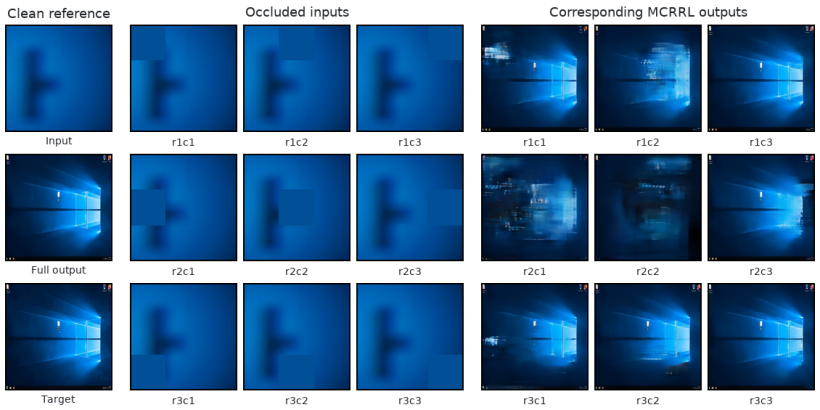

Figure S35: One Screen example under nine-grid occlusion.

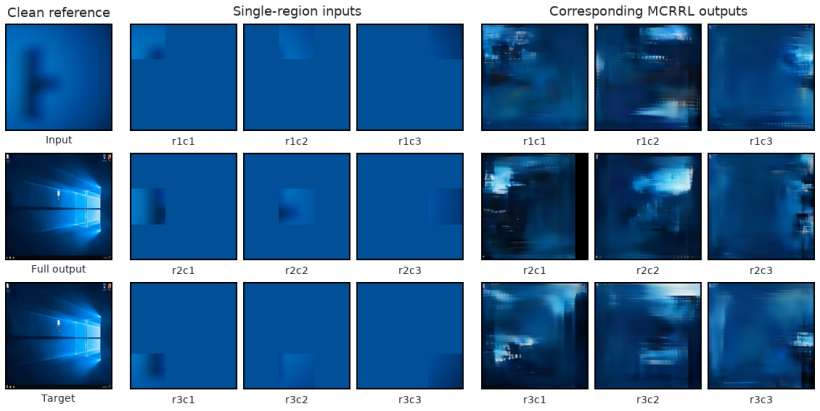

Figure S36: One Screen example under nine-grid single-region retention.

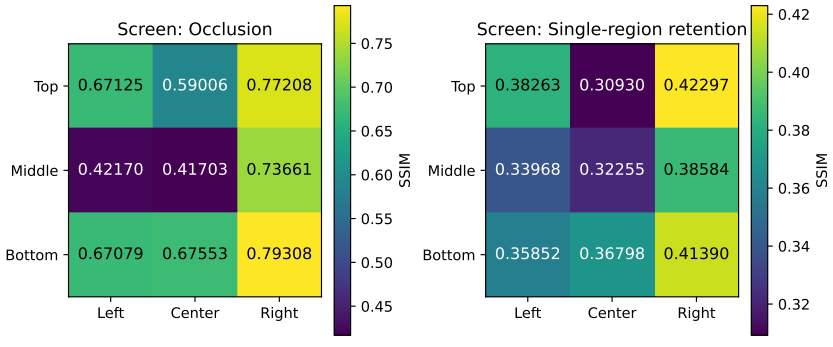

Figure S37: Complete-split SSIM heatmaps for nine-grid occlusion and single-region retention on Screen.

two maps therefore reveal different aspects of the reconstruction: dependence on a region within the complete observation and the recovery supported by that region alone.

C.13 Stage Response and Spectral Mechanism Diagnostics

Figure S38 makes the path from diffuse wall cues to screen structure visible inside the network. The early responses follow the observed wall-light pattern; the decoder and reconstruction stages develop responses around the foreground organization of the target screen.

We display the channel-wise absolute mean of each feature tensor using its own 2nd–98th percentile range and a shared colormap. These stage-wise maps connect the representation and composition steps to the structures present in the final prediction.

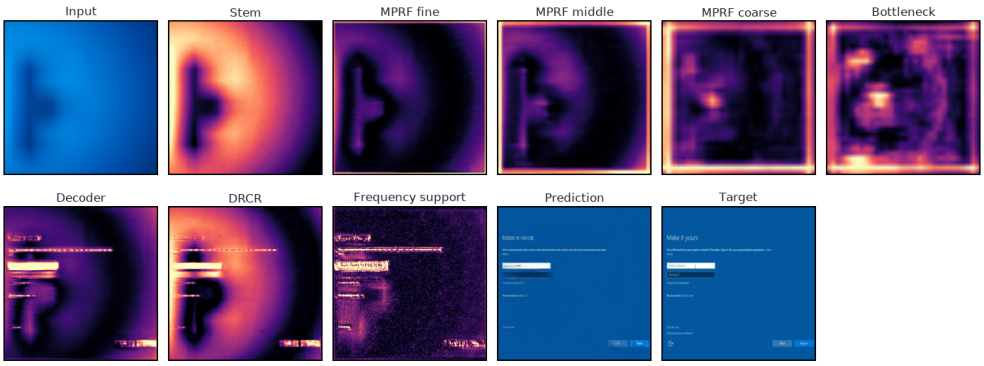

Figure S38: Relative feature responses from the stem through MPRF, decoder, DRCR, and final readout on one Screen example. Each layer is normalized independently for visualization.

Supplementary Figure S39 compares complete-system and deletion-variant responses under the listed spectral perturbations.

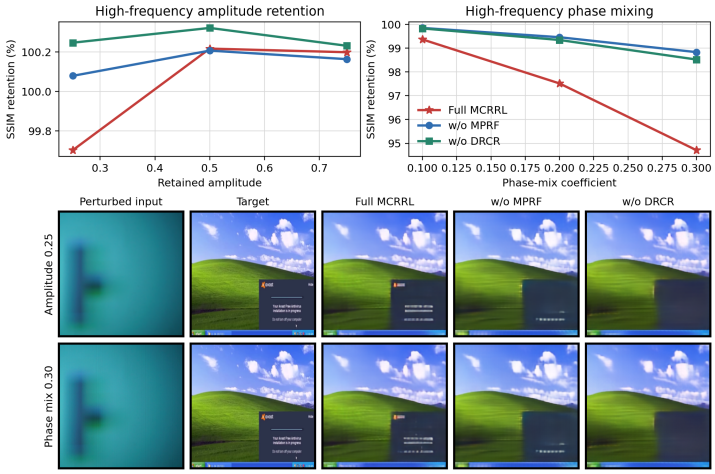

Figure S39: Screen reconstruction under high-frequency amplitude attenuation and phase mixing. The curves show SSIM relative to each variant's clean reference, and the images show the corresponding foreground and layout recovery. Across amplitude-retention factors of 0.25–0.75, SSIM retention is 99.7–100.3%. At a phase-mixing coefficient of 0.30, retention is approximately 94.7%, 98.8%, and 98.5% for the complete model, the model without MPRF, and the model without DRCR, respectively.

The top curves plot the SSIM retention rate against the amplitude retention and the phase mixing coefficient, with real-valued projection after mixing, computed from scores aggregated over the complete Screen test set with fixed model parameters. The bottom rows display the perturbed input, the target, and the outputs of the complete model, the model without MPRF, and the model without DRCR on a fixed sample.

The foreground dialog remains recoverable under high-frequency wall perturbations. Amplitude attenuation produces little change in normalized SSIM across the three variants. Under phase mixing, the complete reconstruction preserves the dialog and visible stroke groups more clearly than the deletion outputs. The curves and images thus connect spectral response to recovered screen content.