Learning Where Light Goes: Transport Routing and Counterfactual Recovery for Hidden Screen Reconstruction

Anonymous authors
Paper under double-blind review

ABSTRACT

Can an ordinary RGB camera recover the content of a hidden screen from the light it casts onto a wall? Diffuse propagation spreads each screen region across multiple measurement locations and mixes distant contributions, obscuring where weak evidence originates. The resulting ambiguity also allows plausible character strokes, boundaries and textures to enter a reconstruction without sufficient measurement support. We propose Transport Coupling and Counterfactual Recovery Learning (TCCRL) to connect evidence attribution with local structure verification. Screen-Wall Transport Coupling Learning learns a nonnegative sparse coupling shared across contents of each fixed capture geometry. The same coupling predicts measurement effects and routes wall evidence back to an initial screen state. Counterfactual Transport Verification and Selective Write-back then tests bounded local additions and deletions against complementary measurements and sequentially incorporates accepted changes. Across five screen-content benchmarks, TCCRL improves reconstruction fidelity and perceptual quality over 19 baselines. The recovered images preserve principal colours, regional layouts, page structure, chart outlines, character shapes and interface boundaries. These results establish the initial feasibility of hidden screen reconstruction from wall-mediated observations in fixed physical scenes.

1 INTRODUCTION

A display outside the camera's view still leaves a content-dependent pattern on nearby surfaces. Can that pattern reveal what is on the screen? Within the broader development of indirect imaging Faccio et al. (2020), computational periscopy has shown that occluder-shaped wall illumination can reveal a hidden two-dimensional scene using an ordinary digital camera Saunders et al. (2019). We study hidden screen reconstruction from wall-mediated RGB observations, seeking the screen's colours, layout and structural boundaries in this indirect light. Figure 1 summarizes the task scenario, while Appendix Figures 9 and 10 document the two capture arrangements. Within each fixed geometry, we learn how the visible response relates to screen locations and use that correspondence to recover the hidden content.

Two difficulties separate the wall response from the hidden screen. First, each screen region spreads light across multiple measurement locations, while each measurement neighbourhood combines contributions from different screen regions. Diffuse attenuation, static wall appearance and fixed foreground occlusion obscure the origin of the remaining content-dependent response. Computational periscopy and unknown-occluder imaging show why forward-path structure matters Saunders et al. (2019); Yedidia et al. (2019). Second, several local screen structures can explain the same weak observation. A reconstruction prior can supply plausible strokes and boundaries, while the wall evidence must determine which proposed changes should enter the result. Recent inverse-imaging methods combine generative priors with measurement-guided recovery Chung et al. (2023); Cai et al. (2024). This motivates testing local screen hypotheses against evidence beyond the observations used to propose them.

To address these challenges, we propose Transport Coupling and Counterfactual Recovery Learning (TCCRL). Screen-Wall Transport Coupling Learning (SWTCL) learns, for each fixed scene, a reusable nonnegative sparse coupling C_s between screen and wall representations. This coupling predicts how screen features contribute to the wall measurement and maps wall evidence back to the screen, providing a shared correspondence for reconstruction and verification. Aggregating this evidence across scales yields an initial screen estimate. Counterfactual Transport Verification and Selective Write-back (CTVSW) then asks whether a proposed local refinement is supported by the observation. It generates bounded refinements from one observation partition and retains those that improve consistency with complementary measurements under the same coupling. The final image combines the initial and refined reconstructions according to the spatial support of accepted changes. The experiments connect this route-and-verify process to recoverable screen structure: principal colours, regional layouts, chart outlines, character shapes and interface boundaries emerge from wall-mediated observations across the two capture arrangements. Together, these results establish the initial feasibility of hidden screen reconstruction in fixed physical scenes.

The main contributions are summarized as follows.

(1) We establish the initial feasibility of recovering hidden screen content from wall-mediated observations recorded by an ordinary RGB camera in fixed physical scenes.

(2) We introduce a scene-shared transport coupling and complementary counterfactual verification, linking the attribution of wall evidence to the selective write-back of local screen structure.

(3) We evaluate five screen-content benchmarks through reconstruction comparisons, ablations, qualitative analysis, digital perturbations and separately learned physical capture conditions.

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

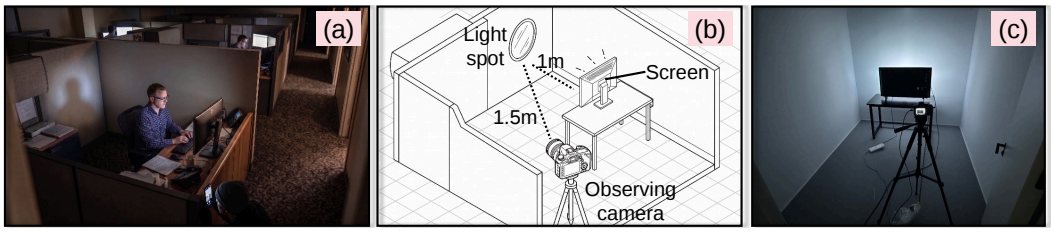


Figure 1: Overview of the task scenario: recovering content on a hidden screen through its reflected light spot on the wall. (a) An exemplar office scene in which the observer sees only the reflected wall spot. (b) Geometry diagram: light emitted by the screen reaches the wall behind the occluder, and the camera records the visible reflected spot without directly observing the screen. (c) A real room arrangement used to collect reflected-light observations.

The two physical capture arrangements are documented in Appendix A; the main text uses these configurations to motivate the scene-shared coupling and the evaluation protocol.

2 Related Work

Indirect imaging makes hidden content recoverable by exploiting how it changes visible measurements. We review transport-aware hidden-scene imaging and learned reconstruction methods that connect image structure to measurement evidence.

2.1 Wall-mediated hidden-scene imaging

Passive wall-mediated reconstruction extracts hidden-scene information from indirect intensity measurements. Early geometric approaches used corner and wall appearance as a camera surrogate Bouman et al. (2017), while confocal time-of-flight methods recovered hidden geometry through the light-cone transform O'Toole et al. (2018). Wave-based formulations later used fast f-k migration for transient measurements Lindell et al. (2019). Saunders et al. proposed a computational periscopy method, which uses occluder-induced transport structure to recover a hidden scene from a wall photograph Saunders et al. (2019). Yedidia et al. proposed an unknown-occluder method, which uses inferred visibility structure to recover hidden scenes Yedidia et al. (2019).

Geng et al. proposed an optimal-transport method, which uses distribution alignment to handle passive reconstruction Geng et al. (2022). Learned NLOS models have also explored neural implicit surfaces, transformer reconstruction, under-scanned transients and virtual scanning Fujimura et al. (2023); Li et al. (2023a;b); Cui et al. (2024); recent scene-specific systems include Flatland and LEAP Peña et al. (2025); Cho et al. (2024). Time-resolved approaches organize indirect measurements through propagation models and learned measurement representations, including confocal reconstruction, wave-based migration and transient recovery. Flatland couples two-dimensional transient simulation with phasor-field imaging Peña et al. (2025). Liu et al. reconstruct hidden three-dimensional structure from sparsely sampled, time-resolved laser measurements, while LEAP recovers clean aperture phasor fields from noisy, incomplete measurements of this kind Liu et al. (2023); Cho et al. (2024). For two-dimensional hidden screens, we learn the relation between screen content and ordinary RGB wall observations, reaching 25.40 dB PSNR on the browser benchmark (Table 2). Zhang et al. proposed a light-transport-modulation method, which uses modulation of indirect transport to improve passive imaging Zhang et al. (2025). Zhang et al. (2025) infer and quantize a transport-condition code from the projection image, then use it to modulate jointly trained reconstruction and reprojection networks. TCCRL learns an explicit sparse correspondence between screen and measurement sites, using the same scene-shared coupling to aggregate evidence and test the measurement effects of local edits before write-back.

2.2 Learned Restoration and Consistency-Aware Readout

This family combines learned image representations with an explicit choice of reconstruction space and data-consistency mechanism. U-Net established a widely reused encoder-decoder template Ronneberger et al. (2015); later restoration backbones include Uformer, HAT, SFHformer, AdaIR and VarFormer Wang et al. (2022); Chen et al. (2023b); Jiang et al. (2024); Cui et al. (2025); Wang et al. (2025). Zamir et al. proposed Restormer, which uses channel-oriented attention to handle high-resolution restoration Zamir et al. (2022). Chen et al. proposed NAFNet, which uses simplified restoration blocks to handle degraded images Chen et al. (2022).

For related degradations, DRSformer, DarkIR, lightweight multi-scale fusion and PhoCoLens target deraining, low-light restoration, passive NLOS imaging and lensless recovery, respectively Chen et al. (2023a); Feijoo et al. (2025); Chen et al. (2025); Cai et al. (2024). VolFormer extends spatial–spectral interaction modeling for hyperspectral image restoration Yu & Gao (2025). Monakhova et al. proposed a lensless reconstruction method, which uses an optical forward model and learned reconstruction to recover scene images Monakhova et al. (2019). Measurement-guided reconstruction also appears in generative inverse imaging. Diffusion Posterior Sampling steers diffusion updates with a measurement-discrepancy gradient Chung et al. (2023), while PhoCoLens combines spatially varying deconvolution with conditional diffusion to recover consistent lensless images Cai et al. (2024). In TCCRL, measurement agreement becomes an acceptance rule for bounded local screen edits: the shared coupling predicts each edit's effect, complementary observations test it, and the accepted support determines its contribution to the final image.

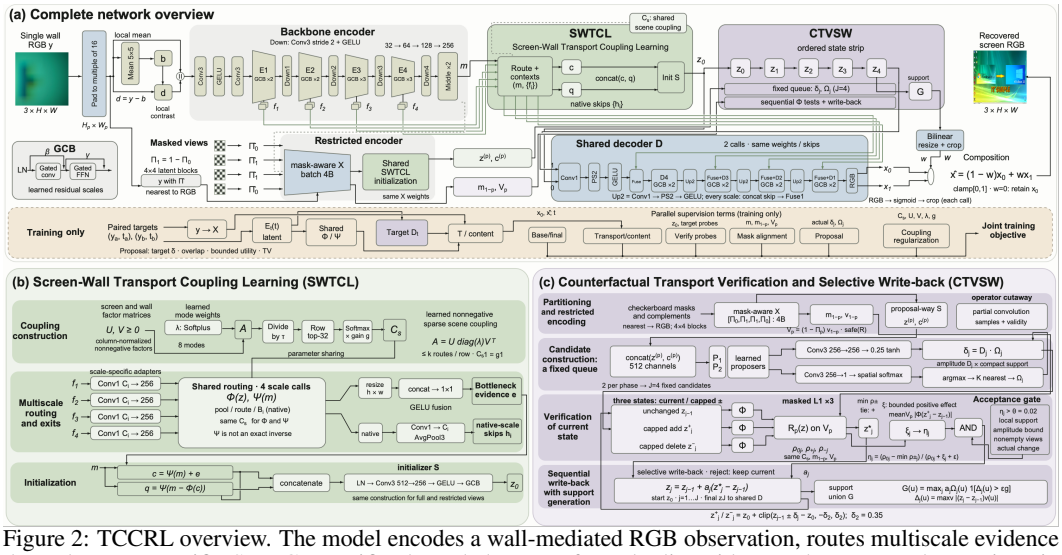


Figure 2: TCCRL overview. The model encodes a wall-mediated RGB observation, routes multiscale evidence through scene-specific SWTCL, verifies bounded counterfactual edits with complementary observations in CTVSW, and combines the initial and selectively refined reconstructions. The detailed training-only terms are shown for completeness.

3 METHOD

TCCRL links two decisions: where wall evidence should be assigned, and which screen changes that evidence supports. SWTCL learns the scene-shared correspondence and uses it to form a base screen state. CTVSW then evaluates bounded local hypotheses against complementary measurements and writes back accepted changes. The same coupling connects both decisions, so the route used to recover content also predicts the evidence required to support its refinement. Figure 2 summarizes the reconstruction flow, and Appendix B details the backbone, transport formulation and verification rules.

3.1 SCREEN-WALL TRANSPORT COUPLING LEARNING

Diffuse transport mixes screen contributions with static scene appearance. We encode the observation y together with its local contrast d into a measurement field $m \in \mathbb{R}^{C\times h\times w}$ and multiscale features f_i:

$$b(u) = \frac{1}{|\mathcal{N}(u)|}\sum_{v\in\mathcal{N}(u)} y(v), \qquad d = y - b, \qquad (m, \{f_i\}_{i=1}^{L}) = \mathcal{X}(y, d). \tag{1}$$

Here, $\mathcal{N}(u)$ is a fixed neighbourhood; batch indices are omitted. Local centring exposes spatial contrast, while the original observation preserves absolute colour and intensity. Together, they provide complementary cues for locating content-dependent wall responses.

On an $N = P^2$ routing grid, we learn nonnegative factors $U, V \in \mathbb{R}_+^{N\times r}$ with unit-sum columns and mode weights $\lambda \geq 0$. Let T_α contain the k largest affinities in row α. The scene coupling is

$$A_{\alpha\beta} = \sum_{\gamma=1}^{r} U_{\alpha\gamma}\lambda_\gamma V_{\beta\gamma}, \qquad C_{s,\alpha\beta} = g\frac{\mathbf{1}[\beta\in T_\alpha]e^{A_{\alpha\beta}/\tau}}{\sum_{\beta'\in T_\alpha} e^{A_{\alpha\beta'}/\tau}}, \tag{2}$$

where $g, \tau > 0$ control row mass and concentration. Thus $C_s\mathbf{1} = g\mathbf{1}$, with at most k nonzeros per row. The coupling assigns each screen-side site α a sparse set of measurement-side destinations β. It is shared across screen contents in scene s, so changing the page changes the routed features while retaining the learned spatial correspondence. Each row has mass g and at most k nonzero entries; destination masses vary with the aggregation of screen contributions. Only the affinity A is constrained to rank at most r.

Let A pool each channel onto the routing grid and B bilinearly resample it to the latent lattice. Using spatial column vectors, forward transport and reverse evidence routing are

$$\Phi = \mathsf{B}C_s^\top\mathsf{A}, \qquad \Psi = \mathsf{B}C_s\mathsf{A}, \qquad e = \mathcal{E}\big(\{\Psi(\mathcal{E}_i(f_i))\}_{i=1}^{L}\big). \tag{3}$$

The learned maps $\mathcal{E}_i, \mathcal{E}$ align and aggregate multiscale evidence. Forward prediction and reverse aggregation use the same coupling with opposite index orientations. A screen feature is therefore linked to the measurement effect it predicts and to the wall evidence assigned back to that location. Routed evidence and unexplained measurements initialize the screen state:

$$c = \Psi(m) + e, \qquad q = \Psi(m - \Phi(c)), \qquad z_0 = \mathcal{S}(c, q). \tag{4}$$

The initializer combines two views of the same evidence: the response already assigned to screen locations and the measurement response still unexplained by that assignment. Multiscale fusion adds layout context and boundary cues, turning these complementary signals into the base screen state. Multiscale encoders preserve coarse layout and fine boundary cues; the shared route is applied at each scale, and the learned fusion maps bring them to a common latent lattice before initialization.

The two orientations of C_s serve related but distinct computations. In Φ, features pooled from the screen lattice are distributed to measurement locations to predict their response; in Ψ, observed features are accumulated at screen locations to form reconstruction evidence. Since the same sparse matrix governs both operations, these maps keep the initial estimate tied to measurement effects that the model can later test. This shared parameterization also makes the two operations mutually constraining: a route that gathers evidence at a screen location must correspond to a forward prediction of where that location affects the wall. The learned coupling is a scene-specific feature correspondence rather than a calibrated optical kernel: it captures reusable spatial associations in the training pairs, while the learned encoders and fusion maps provide the feature representation in which that association is applied. A different fixed geometry changes these associations and is represented by its own C_s.

The same route must explain both a screen and the changes between screens. Individual-sample consistency links screen features to wall responses; content differences reduce the influence of components that remain fixed across contents; convex-mixture consistency further couples these responses under the shared route. Reconstruction supervision supplies the screen-image prior, while these constraints teach it to follow content-dependent wall evidence. Appendix B.5 specifies the training objective and settings. A new physical geometry requires a new scene coupling.

3.2 Counterfactual Transport Verification and Selective Write-back

Partitioned proposals. To assess a refinement using evidence outside the observations that generated it, we divide the observation into complementary spatial partitions. Block-checkerboard masks $\Pi_0, \Pi_1 = 1 - \Pi_0$ define the partitions on the latent lattice Λ. Each partition is encoded using only its own input samples, so local feature aggregation cannot mix evidence across partitions. Applying Eq. (4) separately yields states $z^{(p)}$ and contexts $c^{(p)}$. These fields generate J local candidates before refinement begins, with $\|\delta_j\|_\infty \le \delta_1$ and spatial support Ω_j around a learned location. Verification uses the complementary partition and excludes a neighbourhood of radius R around the proposal observations:

$$V_p(u) = (1 - \Pi_p(u))v_{1-p}(u)\mathbf{1}[\mathrm{dist}_\infty(u, \mathcal{P}_p) > R], \qquad \mathcal{P}_p = \{u : \Pi_p(u) = 1\}. \tag{5}$$

Here v_{1-p} identifies complementary features supported by observed samples. A proposal is formed from one input partition, and its predicted effect is evaluated on the complementary partition. Mask-aware encoding preserves this separation during feature extraction, while the exclusion region separates proposal sites from nearby verification sites. The full observation supplies the base state z_0; the partitioned states supply the local hypotheses. Appendix B specifies partition encoding and candidate support construction.

Counterfactual verification. Each candidate is tested in both directions to determine whether adding or removing its proposed structure better explains the complementary measurement. The cumulative change is bounded around z_0 to limit drift. Let $\langle F \rangle_{V_p}$ denote the average over channels and verification sites:

$$z_j^{\pm} = z_0 + \mathrm{clip}(z_{j-1} \pm \delta_j - z_0, -\delta_2, \delta_2), \qquad \mathcal{R}_p(z) = \langle |m_{1-p} - \Phi z| \rangle_{V_p}. \tag{6}$$

Let $\rho_{0j} = \mathcal{R}_p(z_{j-1})$, $\rho_{\pm j} = \mathcal{R}_p(z_j^{\pm})$, and choose $z_j^\star$ with the smaller residual. We express improvement relative to the current discrepancy and a reference transport effect ξ_j, making the score dimensionless. The bounded positive update defines this reference before direction selection:

$$\xi_j = \langle |\Phi(z_j^+ - z_{j-1})| \rangle_{V_p}, \qquad \eta_j = \frac{\rho_{0j} - \min(\rho_{+j}, \rho_{-j})}{\rho_{0j} + \xi_j + \epsilon}. \tag{7}$$

Here $\epsilon > 0$ stabilizes normalization. A candidate earns write-back by reducing the complementary residual relative to the current state. The score scales this reduction by the current discrepancy and the candidate's reference transport effect. Acceptance also requires valid local support and a nonzero bounded change. Appendix B.4 gives the exact checks and numerical conventions. Accepted candidates update the state sequentially:

$$a_j = \mathbf{1}[\eta_j > \theta]\mathbf{1}[\mathcal{A}_j], \qquad z_j = z_{j-1} + a_j(z_j^\star - z_{j-1}), \qquad \theta \ge 0. \tag{8}$$

Every accepted edit improves agreement on its associated verification mask and keeps the cumulative state change within δ_2. Sequential acceptance therefore combines local measurement improvement with bounded refinement. Training uses straight-through surrogates for location and acceptance, retaining hard forward decisions.

Evaluating both signs makes the update directional: the method can add or suppress a proposed feature according to which bounded state better explains the complementary observation. The reference effect ξ_j places the reduction on a scale set by the candidate's predicted measurement change, while the current residual in the denominator accounts for the remaining mismatch. The score is used as a relative acceptance criterion within the learned feature and transport space. The cumulative bound is centred on z_0, so sequential updates adapt to earlier accepted edits without allowing their total displacement to grow without limit.

Support-gated readout. The refined reconstruction contributes only where verification supports a change; elsewhere, the initial reconstruction is retained. Let G mark the latent sites changed by accepted candidates. Bilinear interpolation B^x maps this support to image coordinates:

$$w = \mathsf{B}^x G. \tag{9}$$

Both states use the same decoder and multiscale evidence, so their reconstruction difference is induced by the accepted latent updates. The support gate then determines where that refinement contributes to the final screen. Their outputs are combined according to w:

$$x_0 = \mathcal{D}(z_0, \{h_i\}), \qquad x_1 = \mathcal{D}(z_J, \{h_i\}), \qquad \widehat{x} = (1 - w)x_0 + wx_1. \tag{10}$$

Since $w \in [0, 1]$, pixels with zero gate retain the base reconstruction. The final image thus combines the base screen with selectively verified local changes, using fixed learned maps and the scene coupling at inference.

4 Experiments and Results

4.1 Experimental Settings

We reconstruct the hidden screen from the wall-mediated RGB observation acquired in a fixed physical scene. Paired observations and screen images supervise training, and the corresponding test targets provide the reconstruction

Table 2: Comparison with 19 baselines on five benchmarks: PSNR (dB)↑, SSIM↑ and LPIPS↓. Source lists publication venue and year (IEEE TIP: IEEE Transactions on Image Processing; IEEE TCI: IEEE Transactions on Computational Imaging). Red/blue mark the best/second best values per column; matching row shading marks rows with either rank.

Method	Source	CtM			SmM			browser			char			software		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Methods for Hidden-Scene Imaging																
Flatland	IEEE TCI 2025	16.80	0.6895	0.3491	21.21	0.7312	0.3203	19.28	0.7080	0.3359	12.57	0.7952	0.1694	23.25	0.8249	0.2456
LEAP	ECCV 2024	17.43	0.7060	0.3053	24.06	0.7957	0.2061	22.52	0.8037	0.2012	17.20	0.8874	0.1022	22.86	0.8267	0.1850
NLOSNeuS	ICCV 2023	14.03	0.6375	0.4638	14.76	0.5725	0.4780	13.59	0.5875	0.4897	9.50	0.7975	0.2354	15.17	0.6879	0.4091
NLOST	CVPR 2023	15.41	0.6434	0.4920	14.14	0.5386	0.6213	14.92	0.5633	0.5806	10.10	0.7939	0.3332	16.83	0.6859	0.4986
NLOS-OT	IEEE TIP 2022	17.35	0.7033	0.3099	23.33	0.7837	0.2238	22.82	0.8078	0.1945	14.75	0.8352	0.1331	22.30	0.8213	0.1970
Under-scanning	NeurIPS 2023	15.23	0.6449	0.4771	14.90	0.5707	0.5636	15.21	0.5709	0.5495	11.59	0.7887	0.2163	17.03	0.6920	0.4706
Virtual scanning	NeurIPS 2024	17.66	0.7174	0.3109	21.50	0.7358	0.2977	16.83	0.6649	0.3616	12.98	0.8262	0.1490	20.05	0.7886	0.2927
General-Purpose Image Restoration																
AdaIR	ICLR 2025	16.29	0.6681	0.3631	22.22	0.7457	0.2548	19.84	0.7451	0.2502	13.43	0.8269	0.1460	19.58	0.7628	0.3213
HAT	CVPR 2023	14.40	0.6407	0.4721	13.60	0.5363	0.5336	13.12	0.5830	0.5207	9.41	0.7992	0.2232	14.04	0.6497	0.4539
NAFNet	ECCV 2022	17.56	0.7111	0.2987	23.65	0.7905	0.2155	21.67	0.7850	0.2165	16.29	0.8594	0.1137	23.11	0.8272	0.1918
Restormer	CVPR 2022	17.56	0.7063	0.3032	23.85	0.7984	0.2093	23.06	0.8086	0.1917	17.87	0.8814	0.0922	22.79	0.8180	0.1813
SFHformer	ECCV 2024	17.35	0.7015	0.3728	19.82	0.6862	0.3767	18.01	0.6804	0.3910	12.76	0.8246	0.1786	21.79	0.8097	0.2879
U-Net	MICCAI 2015	15.15	0.6363	0.4992	12.70	0.5010	0.6251	12.35	0.5382	0.6172	13.23	0.8330	0.1726	14.53	0.6726	0.4574
Uformer	CVPR 2022	16.65	0.6748	0.3533	22.84	0.7627	0.2413	20.90	0.7679	0.2311	14.19	0.8435	0.1374	20.51	0.7849	0.2774
VarFormer	CVPR 2025	14.76	0.6443	0.4838	14.44	0.5591	0.5601	13.72	0.5747	0.5554	10.22	0.7905	0.3076	15.27	0.6770	0.4955
Models for Related Degradations																
DarkIR	CVPR 2025	17.25	0.6984	0.3174	23.06	0.7576	0.2373	22.00	0.7931	0.2107	15.41	0.8542	0.1237	22.06	0.8080	0.2054
DRSformer	CVPR 2023	17.72	0.7264	0.3136	23.14	0.7799	0.2349	17.11	0.6783	0.3437	16.26	0.8827	0.1092	20.74	0.7988	0.2537
PhoCoLens	NeurIPS 2024	16.62	0.6752	0.3552	22.47	0.7624	0.2514	20.40	0.7474	0.2441	13.98	0.8166	0.1441	20.33	0.7816	0.2904
VolFormer	CVPR 2025	13.35	0.6378	0.4681	12.39	0.5224	0.5682	10.48	0.4695	0.6114	9.43	0.7969	0.2734	10.13	0.5670	0.5448
Ours (TCCRL)	This work	17.98	0.7313	0.2773	26.19	0.8359	0.1540	25.40	0.8575	0.1531	20.04	0.9212	0.0676	27.84	0.8864	0.1143

reference. Methods use the same data split and evaluation processing. Training uses fixed (256×256) centre crops. Evaluation uses the image extent listed in Table 1, with input padding removed before scoring.

Table 1: Dataset statistics. Counts denote paired samples, each consisting of one input and one target image. The split sizes follow an approximately $8:1:1$ train–validation–test ratio; integer rounding is used when the total is not divisible by ten.

Dataset	Total	Train	Val	Test	Input / target resolution
CtM	7,000	5,600	700	700	256×256 / 256×256
SmM	2,496	1,997	249	250	256×256 / 256×256
char	3,052	2,442	305	305	256×256 / 256×256
software	4,500	3,600	450	450	512×288 / 512×288
browser	4,500	3,600	450	450	512×288 / 512×288

We use five benchmarks that cover distinct screen-content regimes: *CtM* (charts and tables), *SmM* (mixed screen and natural content), *browser* (web pages), *char* (character content) and *software* (software interfaces). Every benchmark is split into training, validation and test sets in an $8:1:1$ ratio using a fixed, seed-controlled assignment at the sample level. Each observation is paired with its corresponding screen image. The benchmarks cover two capture arrangements, documented with representative schematics and photographs in Appendix A (Figures 9 and 10). In Scene A, a T-shaped occluder placed in front of a self-emitting display casts a screen-dependent shadow onto a diffuse white wall; a side-view camera records the wall shadow. In Scene B, the display faces the wall and produces a wall light spot, while a camera behind the display records a composite view containing the rear housing and the surrounding wall. In both arrangements, the emitting screen surface lies outside the camera's view. The housing and occluder belong to the fixed scene geometry, while content-dependent responses appear on the visible relay wall. In Scene B, the housing hides part of that wall, concentrating the available response around the display's outer edges. Each fixed capture geometry is assigned a scene index s and a coupling C_s. The coupling is reused across screen contents acquired in that geometry, giving their wall responses a common spatial correspondence.

We compare nineteen methods in three families. *Methods for hidden-scene imaging* (Flatland, LEAP, NLOSNeuS, NLOST, NLOS-OT, Under-scanning and Virtual scanning) represent propagation-aware or NLOS reconstruction designs. *General-purpose image restoration* (AdaIR, HAT, NAFNet, Restormer, SFHformer, U-Net, Uformer and VarFormer) supplies learned restoration backbones. *Models for related degradations* (DarkIR, DRSformer, PhoCoLens and VolFormer) contribute designs developed for low-light restoration, deraining, lensless imaging and hyperspectral restoration, respectively. Together, the three groups compare different reconstruction designs on the wall-mediated hidden-screen task. We compare all methods on the paired wall-observation and screen-target data. Each implementation maps the wall observation to a screen reconstruction; Appendix B.5 specifies the task adaptations.

We report peak signal-to-noise ratio (PSNR, dB) and structural similarity (SSIM) for fidelity, and VGG-based learned perceptual image patch similarity (LPIPS) for perceptual quality. Higher is better for PSNR and SSIM, lower for LPIPS. All three are computed on the screen image, at the original image extent, after removing input padding. We use the same VGG-based LPIPS implementation for all methods.

4.2 Comparison with Baselines

The wall-mediated observations support reconstruction gains across all five screen-content benchmarks. TCCRL leads all fifteen PSNR, SSIM and LPIPS comparisons in Table 2. The per-metric margins over the strongest baseline of

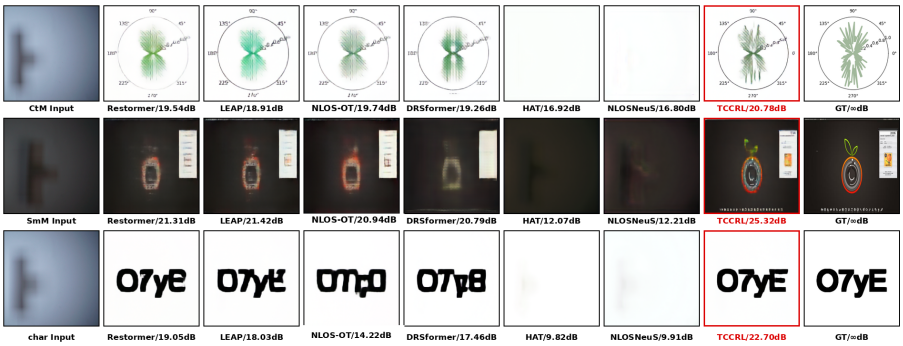


Figure 3: Qualitative comparison in Scene A: side-view wall observations. Rows show *CtM*, *SmM* and *char*, respectively. The camera records screen-induced illumination on the relay wall from the side; the emitting screen surface is outside its direct view. Columns show the input, six baseline reconstructions, TCCRL (red border), and the ground-truth screen image (GT). Values below reconstructions are per-example PSNR in dB (higher is better); ∞ under GT denotes self-comparison.

each column are $+0.26$ dB on CtM, $+2.13$ dB on SmM, $+2.34$ dB on browser, $+2.17$ dB on char and $+4.59$ dB on software, with corresponding SSIM and LPIPS gaps pointing the same way, for example LPIPS 0.1540 against LEAP's 0.2061 on SmM and 0.1143 against Restormer's 0.1813 on software. Averaged over the five benchmarks the margin is $+2.30$ dB PSNR, $+0.037$ SSIM and -0.041 LPIPS. The agreement across pixel fidelity, structural similarity and perceptual quality shows that the gain carries through different measures of the recovered screen. The ablations below then ask which parts of evidence routing and verification produce that gain.

The strongest comparisons come from both hidden-scene imaging and general restoration. LEAP learns informative aperture-field representations for time-resolved imaging (Cho et al., 2024), while Restormer provides an effective high-resolution restoration architecture (Zamir et al., 2022). In our RGB screen-reconstruction evaluation, LEAP spans 17.20–24.06 dB, whereas VolFormer, originating from hyperspectral restoration, spans 9.43–13.35 dB. Restormer reaches 23.06 dB on browser and 17.87 dB on char and is the most frequent runner-up within the general-restoration group. TCCRL improves the screen reconstructions across all three metrics, connecting the comparison to the joint task of assigning diffuse wall evidence and controlling local refinement.

The largest gains occur on screen contents that combine large regions with localized structure. TCCRL improves PSNR by 4.59 dB on software and 2.34 dB on browser, where interface boundaries and small graphical elements must remain distinct from broad uniform areas. On browser, LPIPS also decreases from Restormer's 0.1917 to 0.1531. CtM gains 0.26 dB together with improvements in SSIM and LPIPS, extending the result to chart content. These comparisons connect the quantitative advantage to the central recovery task: preserving the arrangement of screen regions and the structures that separate them.

Figures 3 and 4 show how wall observations translate into recovered screen structure. In Scene A, Figure 3 shows the polar-chart outline and radial pattern, the central graphic against its dark background, and the main character shapes. These examples connect diffuse wall responses to distinct screen regions and their boundaries. In Scene B, the camera observes the display housing and the surrounding wall response; Figure 4 shows recovery of the laptop outline, facial structure and interface boundaries from that indirect evidence. The two arrangements therefore reveal complementary forms of recoverable screen content despite their different visible wall supports. Per-example PSNR describes the displayed reconstructions, while Table 2 summarizes the test-set comparison. Together, the images and aggregate results establish the initial feasibility of reconstructing screen colours, layouts and structural contours from wall-mediated RGB observations.

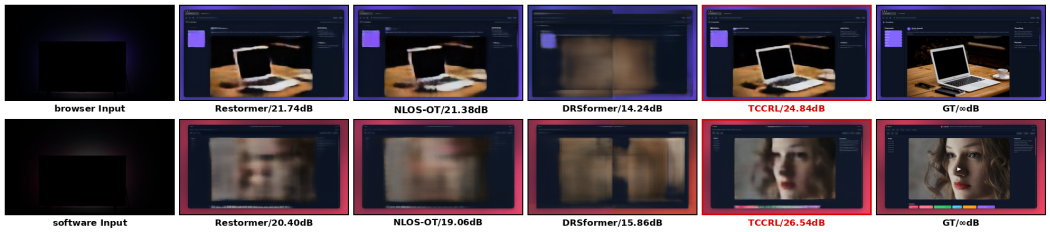


Figure 4: Qualitative comparison in Scene B: rear-view observations with fixed foreground occlusion. Rows show *browser* and *software*, respectively. The camera sees the display's rear housing and the surrounding wall, but not its emitting surface. The housing occludes part of the relay wall; content-dependent light responses lie mainly on the visible wall around its edges. Columns show the input, three baseline reconstructions, TCCRL (red border), and the ground-truth screen image (GT). Values below reconstructions are per-example PSNR in dB; ∞ under GT denotes self-comparison.

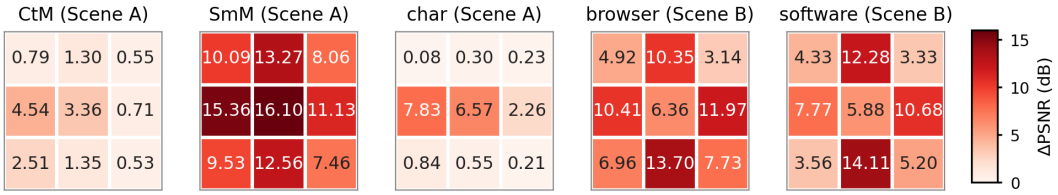


Figure 5: Spatial attribution of screen-dependent wall evidence across five benchmarks. Each panel is a 3×3 input grid; one cell at a time is replaced by its cross-sample mean, and the value is the resulting ΔPSNR (dB). Larger values indicate greater dependence of the reconstruction on that cell's content-dependent observation. The shared colour bar is placed at the right.

4.3 Ablation Study

Table 3: Ablations on *SmM* and *software*. Each cell shows the metric value with its signed difference from Full as a subscript, Δ = variant − Full. Differences are in dB for PSNR and absolute units for SSIM and LPIPS. Red/blue marks the best/second-best value in each column. Full matches TCCRL in Table 2.

Variant	*SmM*			*software*		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Evidence routing (SWTCL)						
w/o local contrast	$25.35_{-0.84}$	$0.8120_{-0.0239}$	$0.1789_{+0.0249}$	$26.85_{-0.99}$	$0.8650_{-0.0214}$	$0.1418_{+0.0275}$
w/o learned route	$24.65_{-1.54}$	$0.7930_{-0.0429}$	$0.2057_{+0.0517}$	$26.00_{-1.84}$	$0.8420_{-0.0444}$	$0.1706_{+0.0563}$
uniform routing	$25.55_{-0.64}$	$0.8160_{-0.0199}$	$0.1753_{+0.0213}$	$26.95_{-0.89}$	$0.8700_{-0.0164}$	$0.1391_{+0.0248}$
input-adaptive routing	$25.88_{-0.31}$	$0.8280_{-0.0079}$	$0.1584_{+0.0044}$	$27.48_{-0.36}$	$0.8830_{-0.0034}$	$0.1171_{+0.0028}$
single-scale evidence	$25.62_{-0.57}$	$0.8190_{-0.0169}$	$0.1712_{+0.0172}$	$27.15_{-0.69}$	$0.8750_{-0.0114}$	$0.1324_{+0.0181}$
Candidate verification and updates (CTVSW)						
w/o CTVSW	$24.85_{-1.34}$	$0.8010_{-0.0349}$	$0.1948_{+0.0408}$	$26.25_{-1.59}$	$0.8510_{-0.0354}$	$0.1619_{+0.0476}$
w/o complement check	$25.10_{-1.09}$	$0.8060_{-0.0299}$	$0.1876_{+0.0336}$	$26.60_{-1.24}$	$0.8590_{-0.0274}$	$0.1517_{+0.0374}$
parallel candidate updates	$25.78_{-0.41}$	$0.8230_{-0.0129}$	$0.1649_{+0.0109}$	$27.30_{-0.54}$	$0.8790_{-0.0074}$	$0.1248_{+0.0105}$
Full TCCRL	$26.19_{0.00}$	$0.8359_{0.0000}$	$0.1540_{0.0000}$	$27.84_{0.00}$	$0.8864_{0.0000}$	$0.1143_{0.0000}$

Table 3 examines evidence routing and candidate verification on *SmM* and *software*. Each entry reports the reconstruction metric and its signed difference from Full, allowing the effect of each variant to be compared within a dataset. Negative PSNR/SSIM differences and positive LPIPS differences indicate degradation. Full achieves the best value in all six columns.

Losing the learned screen–wall correspondence produces the largest ablation penalty. Removing the learned transport coupling drops PSNR by 1.54 dB on SmM and 1.84 dB on software; SSIM falls by 0.0429 and 0.0444, while LPIPS increases by 0.0517 and 0.0563. Removing the local-variation input reduces PSNR by 0.84 and 0.99 dB; uniform routing reduces it by 0.64 and 0.89 dB. Uniform routing tests the value of learned spatial allocation through its routing weights and support. Input-adaptive routing is the strongest alternative in all six columns, trailing the scene-shared Full model by 0.31 and 0.36 dB. Single-scale evidence incurs larger PSNR losses of 0.57 and 0.69 dB, with SSIM and LPIPS also worsening on both datasets. The scene-shared route improves over input-adaptive routing, while multiscale evidence adds useful spatial context. Together, the variants support reusing the scene correspondence and aggregating content-dependent evidence across scales.

Finding the evidence is followed by deciding which changes it supports. Removing CTVSW reduces PSNR by 1.34 dB on SmM and 1.59 dB on software, with LPIPS increasing by 0.0408 and 0.0476. Within this stage, removing complementary verification incurs larger PSNR losses, 1.09 and 1.24 dB, than replacing sequential write-back with parallel updates, which costs 0.41 and 0.54 dB. SSIM and LPIPS show the same ordering. Complementary measurements therefore supply the larger contribution, while sequential write-back adds a further benefit by evaluating each candidate against the state produced by earlier accepted changes.

4.4 Spatial Attribution of Wall Evidence

We ask which parts of the observation the trained reconstruction actually reads. We partition each input into a 3×3 grid and replace one cell at a time by its cross-sample mean, preserving its static background while removing the variation associated with screen content. Figure 5 shows the resulting per-cell ΔPSNR for all five benchmarks; larger values mean that removing the cell's content-dependent component changes the reconstruction more. Scene A concentrates its strongest effects in cells 4 and 5 for *CtM* and *char*, while *SmM* responds broadly. In Scene B, surrounding wall cells carry more useful evidence than the central display-housing cell.

The Scene-B pattern is especially clear in Figure 5. For *browser*, mean-filling cells 2, 4, 6 and 8 lowers PSNR by 10.35, 10.41, 11.97 and 13.70 dB; for *software*, cells 2, 6 and 8 produce drops of 12.28, 10.68 and 14.11 dB. Cell 5, which mainly contains the display housing, costs 6.36 and 5.88 dB respectively. The contrast between a varying housing observation and the greater effect of masking the surrounding wall show that raw input variation alone does not determine what the model uses: its scene correspondence selects the more predictive wall regions.

The experiment distinguishes content-dependent evidence from static appearance. Mean fill replaces a cell with its own validation-input mean, so the manipulation retains its average brightness and background statistics while removing sample-specific variation. This is more targeted than black or noise fill, which also alters static appearance. Same-size random masks account for the generic cost of removing one ninth of the observation. Together, the five panels in Figure 5 show that the reconstruction relies on spatially structured evidence: central shadow regions in Scene A, a broad response for mixed content, and the wall bordering the display in Scene B. Appendix C.7 provides the full protocol, matched-control results and variance-normalized analysis.

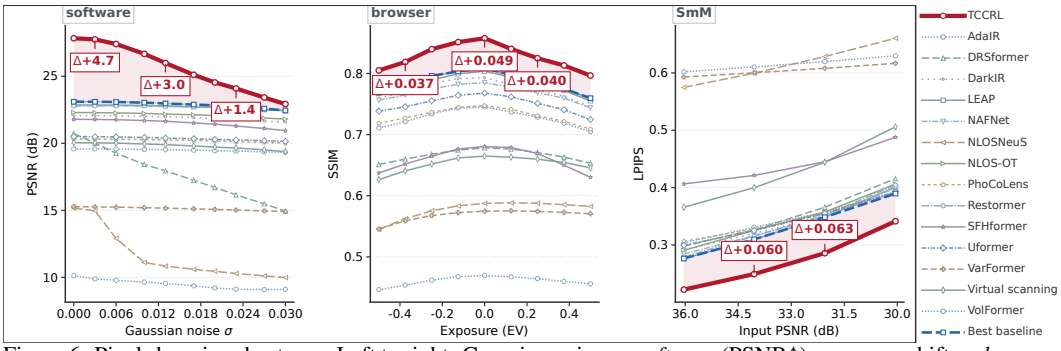


Figure 6: Pixel-domain robustness. Left to right: Gaussian noise on *software* (PSNR↑), exposure shift on *browser* (SSIM↑), and multiplicative speckle on *SmM* (LPIPS↓; severity measured by input PSNR). Red: TCCRL; dashed blue: the best of 14 plotted baselines at each level. Shading and Δ show TCCRL's advantage.

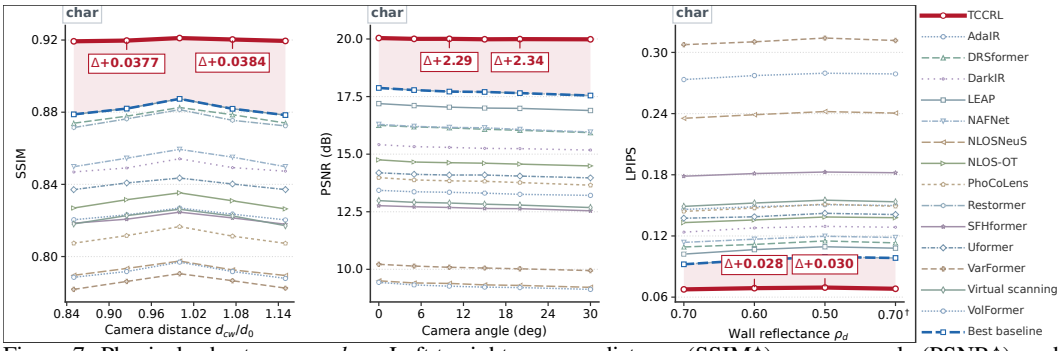


Figure 7: Physical robustness on *char*. Left to right: camera distance (SSIM↑), camera angle (PSNR↑), and wall reflectance (LPIPS↓). Each condition uses a separately trained scene coupling. †: $\rho_d \approx 0.70$ with added surface texture. Red: TCCRL; dashed blue: the best of 14 plotted baselines at each level. Shading and Δ show TCCRL's advantage.

4.5 Robustness Analysis

The clean-image results establish reconstruction ability. We next ask how that ability changes when the recorded wall response is perturbed. TCCRL retains the leading result throughout the three sweeps shown in Figure 6. The figure shows PSNR under additive Gaussian noise on *software*, SSIM under exposure shift on *browser*, and LPIPS under multiplicative speckle on *SmM*. Its margin over the strongest plotted baseline is 1.4 dB at the strongest noise level; under exposure shift, the SSIM margin is 0.037 at -0.5 EV, 0.049 at nominal exposure and 0.040 at $+0.5$ EV; LPIPS remains ahead by at least 0.048 across the four plotted speckle levels. At each level, the blue envelope gives the strongest result among the 14 baselines listed in the legend.

The three perturbations act on different parts of the evidence. Additive noise introduces extra variation, exposure scaling changes the observation's intensity distribution, and speckle adds signal-dependent spatial fluctuations. Complementary verification favours local hypotheses whose predicted effects agree with measurements beyond the proposal partition. Together with the local and cumulative amplitude bounds, this limits how perturbation-driven changes enter the reconstruction. The retained LPIPS advantage under speckle is consistent with this selective-refinement design.

Two choices make the perturbation response directly comparable: severity is expressed through input PSNR, and the trained reconstruction pipeline remains fixed throughout each digital sweep. Speckle is calibrated per benchmark to the common plotted input-PSNR levels, while every perturbation is applied to the already-imaged observation without a corruption-specific model component.

Fig. 7 extends the same question to physical capture geometry on *char*, on three axes and three metrics: SSIM under camera distance d_{cw}/d_0 from 0.85 to 1.15, PSNR under camera angle from 0° to 30°, and LPIPS under wall reflectance ρ_d. TCCRL attains the best value at every level of all three axes as well. The distances and angles are conservative deviations around the original capture pose, measured optically, and each level is a separate scene with its own coupling trained under that geometry, so the figure reports scene-bound stability. The fourth wall condition restores $\rho_d \approx 0.70$ but adds a light fixed surface texture, and TCCRL holds its LPIPS there too, which is the condition that most directly tests the claim that the route follows content-driven variation rather than static wall appearance: texture is precisely the static component that a content-independent route would absorb into the screen estimate.

The digital and physical sweeps probe different failure modes, so their agreement is more informative than either sweep alone. Digital corruption changes the values presented to a fixed scene coupling, whereas a physical change alters the capture geometry and requires a newly learned coupling. The first case tests whether verification can reject edits supported by corrupted measurements; the second tests whether the route-and-verify procedure remains usable after the correspondence itself has been relearned. TCCRL's lead in both settings is therefore consistent with a separation between content- dependent evidence and nuisance variation, while the scene-specific training condition limits the claim to the tested capture family.

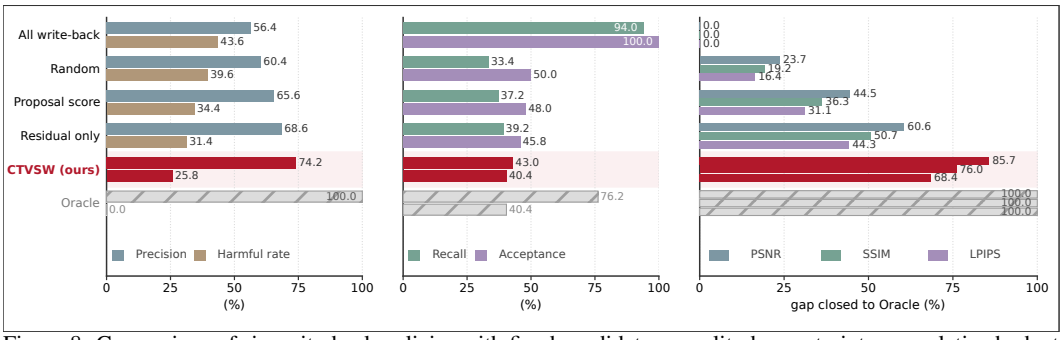


Figure 8: Comparison of six write-back policies with fixed candidates, amplitude constraints, cumulative budget and decoder. Bars: five-benchmark macro-averages. Left: candidate precision and harmful write-back rate. Centre: recall and acceptance rate (fraction written back). Right: PSNR/SSIM/LPIPS gap closed to Oracle, $(\text{policy} - \text{All write-back})/(\text{Oracle} - \text{All write-back})$, with All write-back at 0% and Oracle at 100%. Oracle uses ground truth and matches CTVSW's acceptance count; it is an offline reference.

4.6 Mechanism Analysis: Counterfactual Candidate Selection

The ablations show CTVSW's contribution and motivate testing the mechanism behind that contribution. Fig. 8 isolates the selection step by fixing the candidate set, the amplitude constraints, the cumulative budget and the decoder, and varying only the rule that decides which candidates are written back. Six policies are compared: writing back everything, a random half, ranking by the learned proposal score, ranking by residual improvement alone, CTVSW, and a target-aware oracle that ranks candidates offline with the ground-truth image. The oracle shares CTVSW's write-back budget, so the two accept the same number of candidates and the comparison measures selection quality rather than how much is written back.

Writing back every proposal maximizes acceptance, but candidate quality determines whether those additions improve the screen. Across the five target-free policies, precision rises from 56.4% to 74.2% as the harmful write-back rate falls from 43.6% to 25.8%. The progression links more selective acceptance to cleaner local refinement. CTVSW combines the highest precision with the lowest harmful write-back rate among target-free policies. It accepts 40.4% of candidates, matching the oracle's acceptance budget, while achieving the highest recall, 43.0%, among selective target-free policies.

The decisive comparison concerns which evidence authorizes an edit. Measurement-guided inverse imaging already uses predicted measurement agreement to steer reconstruction (Chung et al., 2023). Here, Figure 8 compares residual-only selection with counterfactual verification on complementary measurements. Precision changes from 68.6% to 74.2%, while the harmful write-back rate changes from 31.4% to 25.8%—a 5.6-percentage-point difference in each measure. Proposal-score selection reaches 65.6% precision. CTVSW evaluates the bounded add and delete states using measurements outside the proposal partition, then tests each accepted change against the updated screen state. The role of verification is explicit: a proposed structure enters the reconstruction after improving its predicted agreement with complementary wall evidence.

The selection difference carries through to the reconstructed image. CTVSW closes 85.7% of the PSNR gap, 76.0% of the SSIM gap and 68.4% of the LPIPS gap between all-write-back and the target-aware oracle. Its five-benchmark macro-averages are 23.4913 dB in PSNR and 0.8465 in SSIM, agreeing with Table 2 at the displayed precision. The oracle supplies a target-aware reference with the same acceptance budget.

The policy comparison also clarifies why acceptance rate alone is an insufficient control variable. All write-back accepts every proposal, but its precision is only 56.4%; a policy can therefore spend the same update budget on edits with very different measurement support. CTVSW's 40.4% acceptance rate is useful because it is matched to the oracle while its candidate precision and recall remain target-free. In this matched-budget setting, the counterfactual check changes which edits consume the budget, rather than simply changing how many edits are allowed.

The mechanism result should consequently be read at two levels. At the local level, complementary measurements reduce the fraction of harmful write-backs. At the image level, sequentially applying the accepted edits allows later candidates to be evaluated against the state that will actually be decoded. The macro-averaged gap closure reports the combined effect of these choices after decoding, and it does not imply that every local candidate is uniquely identified by the wall observation.

5 Conclusion

We demonstrated the initial feasibility of hidden screen reconstruction from wall-mediated light recorded by an ordinary RGB camera. TCCRL connects two complementary decisions: learning where content-dependent wall evidence belongs and testing which local screen changes that evidence supports. SWTCL supplies a scene-shared transport coupling, and CTVSW uses it to verify bounded additions and deletions before selective write-back. Across five screen-content benchmarks, the mean margins over the strongest baseline in each metric–benchmark comparison are 2.30 dB in PSNR, 0.037 in SSIM and 0.041 lower LPIPS. The reconstructions preserve principal colours, regional layouts, page structure, chart outlines, character shapes and interface boundaries. Ablations support the contributions of learned correspondence, complementary verification and sequential updates. Building on ordinary-camera computational periscopy and learned transport-aware reconstruction (Saunders et al., 2019; Zhang et al., 2025), these results establish a route-and-verify approach to recovering hidden screen structure in fixed physical scenes. Learning where the wall evidence goes, and testing the screen changes it supports, turns indirect light into recoverable content. Future work will extend this approach across broader capture geometries and weaker measurement conditions.

REFERENCES

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 *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2289–2297, 2017. doi: 10.1109/ICCV.2017.249.

Xin Cai, Zhiyuan You, Hailong Zhang, Wentao Liu, Jinwei Gu, and Tianfan Xue. Phocolens: Photorealistic and consistent reconstruction in lensless imaging. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 12219–12242. Curran Associates, Inc., 2024. doi: 10.52202/079017-0391. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/16c5b4102a6b6eb061e502ce6736ad8a-Paper-Conference.pdf.

Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *Computer Vision—ECCV 2022*, 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.

Xiang Chen, Hao Li, Mingqiang Li, and Jinshan Pan. Learning a sparse transformer network for effective image deraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5896–5905, June 2023a.

Xiangyu Chen, Xintao Wang, Jiantao Zhou, Yu Qiao, and Chao Dong. Activating more pixels in image super-resolution transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023b. doi: 10.1109/CVPR52729.2023.02142.

In Cho, Hyunbo Shim, and Seon Joo Kim. Learning to enhance aperture phasor field for non-line-of-sight imaging. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

Hyungjin Chung, Jeongsol Kim, Michael T. McCann, Marc L. Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations*, 2023.

Xingyu Cui, Huanjing Yue, Song Li, Xiangjun Yin, Yusen Hou, Yun Meng, Kai Zou, Xiaolong Hu, and Jingyu Yang. Virtual scanning: Unsupervised non-line-of-sight imaging from irregularly undersampled transients. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 109381–109406. Curran Associates, Inc., 2024. doi: 10.52202/079017-3472. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/c58437945392cec01e0c75ff6cef901a-Paper-Conference.pdf.

Yuning Cui, Syed Waqas Zamir, Salman Khan, Alois Knoll, Mubarak Shah, and Fahad Shahbaz Khan. Adair: Adaptive all-in-one image restoration via frequency mining and modulation. In *International Conference on Learning Representations*, 2025.

Daniele Faccio, Andreas Velten, and Gordon Wetzstein. Non-line-of-sight imaging. *Nature Reviews Physics*, 2: 318–327, 2020. doi: 10.1038/s42254-020-0174-8.

Daniel Feijoo, Juan C. Benito, Alvaro Garcia, and Marcos V. Conde. Darkir: Robust low-light image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10879–10889, June 2025.

Yuki Fujimura, Takahiro Kushida, Takuya Funatomi, and Yasuhiro Mukaigawa. Nlos-neus: Non-line-of-sight neural implicit surface. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10532–10541, October 2023.

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.

Xingyu Jiang, Xiuhui Zhang, Ning Gao, and Yue Deng. When fast fourier transform meets transformer for image restoration. In *European Conference on Computer Vision*, pp. 381–402, 2024.

Yue Li, Jiayong Peng, Juntian Ye, Yueyi Zhang, Feihu Xu, and Zhiwei Xiong. NLOST: Non-line-of-sight imaging with transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023a. doi: 10.1109/CVPR52729.2023.01279.

Yue Li, Yueyi Zhang, Juntian Ye, Feihu Xu, and Zhiwei Xiong. Deep non-line-of-sight imaging from under-scanning measurements. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 59095–59106. Curran Associates, Inc., 2023b. doi: 10.52202/075280-2578. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/b91cc0a242e6518ee731f74e82b2eebd-Paper-Conference.pdf.

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.

Xintong Liu, Jianyu Wang, Leping Xiao, Zuoqiang Shi, Xing Fu, and Lingyun Qiu. Non-line-of-sight imaging with arbitrary illumination and detection pattern. *Nature Communications*, 14:3230, 2023. doi: 10.1038/s41467-023-38898-4.

Kristina Monakhova, Joshua Yurtsever, Grace Kuo, Nick Antipa, Kyrollos Yanny, and Laura Waller. Learned reconstructions for practical mask-based lensless imaging. *Optics Express*, 27:28075–28090, 2019. doi: 10.1364/OE.27.028075.

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

María Peña, Diego Gutierrez, and Julio Marco. Looking around flatland: End-to-end 2d real-time nlos imaging. *IEEE Transactions on Computational Imaging*, 11:189–200, 2025. doi: 10.1109/TCI.2025.3536092.

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*, 565:472–475, 2019. doi: 10.1038/s41586-018-0868-6.

Siyang Wang, Naishan Zheng, Jie Huang, and Feng Zhao. Navigating image restoration with var's distribution alignment prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7559–7569, June 2025.

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)*, 2022. doi: 10.1109/CVPR52688.2022.01716.

Adam B. Yedidia, Manel Baradad, Christos Thrampoulidis, William T. Freeman, and Gregory W. Wornell. Using unknown occluders to recover hidden scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. doi: 10.1109/CVPR.2019.01251.

Dabing Yu and Zheng Gao. Volformer: Explore more comprehensive cube interaction for hyperspectral image restoration and beyond. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28091–28101, June 2025.

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 (CVPR)*, 2022. doi: 10.1109/CVPR52688.2022.00564.

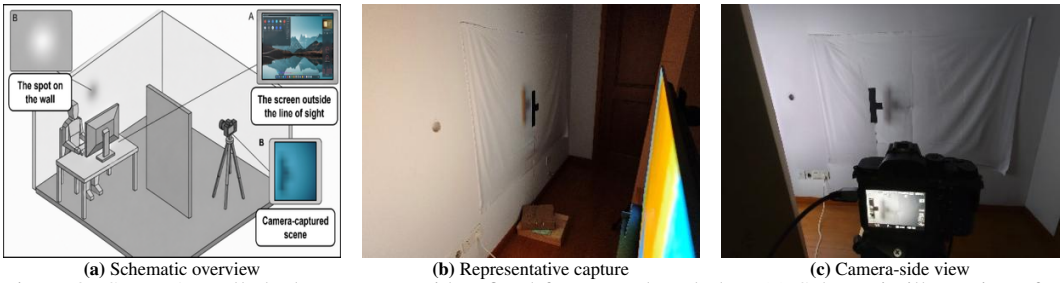
Jiarui Zhang, Ruixu Geng, Xiaolong Du, Yan Chen, Houqiang Li, and Yang Hu. Passive non-line-of-sight imaging with light transport modulation. *IEEE Transactions on Image Processing*, 2025. doi: 10.1109/TIP.2024.3518097.

SUPPLEMENTARY MATERIAL FOR LEARNING WHERE LIGHT GOES: TRANSPORT ROUTING AND COUNTERFACTUAL RECOVERY FOR HIDDEN SCREEN RECONSTRUCTION

This supplement provides further description of the two data-gathering scenes, datasets and formulation derivations for TCCRL. It also reports extended experiments and additional metrics that evaluate recovery under varied noise, exposure, camera distance and camera angle. The scene schematics and photographs in Figures 9 and 10 show the physical layouts behind the two relay-wall observation types used in the experiments.

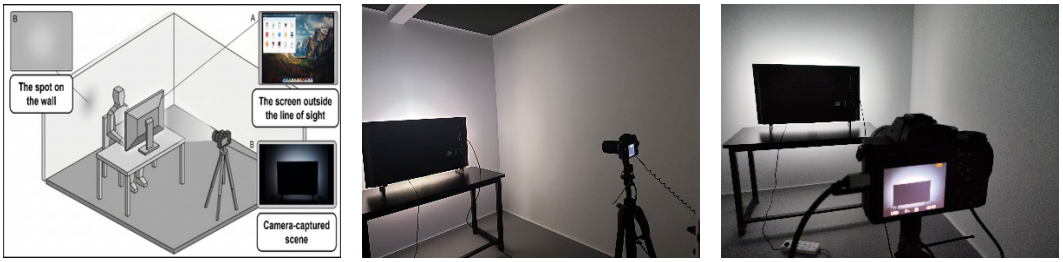
A CAPTURE ARRANGEMENTS

The two capture arrangements provide different visible supports for the screen-dependent light. Scene A uses a foreground occluder to cast a content-dependent shadow onto the relay wall, whereas Scene B records the illuminated wall around the display housing from behind the screen. In both cases the camera observes the relay surface rather than the emitting screen, and the scene geometry remains fixed across screen contents. These schematics and photographs document the physical configurations summarized in Section 4.



(a) Schematic overview **(b)** Representative capture **(c)** Camera-side view

Figure 9: Scene A: wall-shadow capture with a fixed foreground occluder. (a) Schematic illustration of a self-emitting 60-inch LED display, a T-shaped occluder placed about 0.5 m in front of it, the relay wall and the side-view camera; Schematic of the display, T-shaped occluder, relay wall and side-view camera. (b,c) Photographs of the physical arrangement and camera-side view. The screen content is outside the camera's direct line of sight, and the paired observation is the wall image containing the content-dependent shadow.



(a) Schematic overview **(b)** Representative capture **(c)** Camera-side view

Figure 10: Scene B: rear-view capture of a wall light spot. (a) Schematic illustration of a self-emitting 60-inch LED display facing a diffuse wall at about 0.5 m and a Sony A7SII camera placed behind the display; Schematic of the rear-view capture arrangement. (b,c) Photographs of the display housing, wall illumination and camera viewpoint. The observation contains the housing and surrounding wall response, while the emitting surface lies outside the camera's view.

B TRANSPORT AND VERIFICATION DETAILS

This appendix gives the physical motivation and the exact numerical definitions supporting Section 3.

B.1 FEATURE ENCODER AND SHARED RECONSTRUCTION BACKBONE

The backbone converts the RGB observation into the feature fields used for transport and verification. It extracts image structure at several scales, while the scene-shared coupling determines how the corresponding wall evidence is assigned to screen locations. Let the input wall observation be $y \in [0,1]^{B\times 3\times H\times W}$. Before encoding, the implementation pads the bottom and right sides to a multiple of 16 (reflection padding when valid, replication for very small inputs); all outputs and gates are cropped back to the original $H \times W$ extent. The measurement stem computes a reflected 5×5 local average b, forms the differential input $d = y - b$, concatenates y and d, and applies two 3×3 convolutions with a GELU between them. This representation preserves absolute appearance and local variation for the shared encoder.

The encoder has four resolution stages with channel widths $(w, 2w, 4w, 8w)$, where the default implementation uses $w = 32$. Each stage applies its configured number of gated convolution blocks and stores a skip feature; a stride-two 3×3 convolution followed by GELU then downsamples to the next stage. The final $8w$-channel representation passes through two additional gated blocks to produce the bottleneck measurement m. Each gated block uses channel-wise layer normalization, a 1×1 expansion to $2C$ channels, a 3×3 depthwise convolution, a channel split and multiplicative

gating, and a 1×1 projection. A second normalized pointwise expansion and multiplicative gate forms the feed-forward branch. The two branches are added residually with learned per-channel scale vectors initialized to zero:

$$\mathcal{G}_1(x) = W_1^p(\mathrm{GELU}(a_1) \odot b_1)\,, \quad (a_1, b_1) = \mathrm{split}\big(W_1^{dw} W_1^e\,\mathrm{LN}(x)\big)\,, \tag{11}$$

$$x' = x + \beta \odot \mathcal{G}_1(x), \qquad \mathcal{G}_2(x') = W_2^p(\mathrm{GELU}(a_2) \odot b_2)\,, \tag{12}$$

$$(a_2, b_2) = \mathrm{split}(W_2^e\,\mathrm{LN}(x'))\,, \qquad \mathrm{GatedBlock}(x) = x' + \gamma \odot \mathcal{G}_2(x'). \tag{13}$$

Here W^e, W^{dw}, W^p denote pointwise expansion, depthwise spatial filtering and pointwise projection, respectively; the first expansion and depthwise operation occur before the first split. The gate multiplies two equal channel groups elementwise, and β, γ are learned channel scales. This backbone supplies representation capacity; the scene coupling, rather than an image-conditioned attention map, determines how its spatial evidence is routed.

For each encoder skip s_i, a learned 1×1 adapter maps channels to the bottleneck width before applying the same wall-to-screen back-projection Ψ. The transported features are resized to the bottleneck lattice, concatenated and fused by a 1×1 convolution and GELU to obtain the multiscale screen context e. Separately, decoder skips are formed by adapting each s_i, applying Ψ, projecting back to that stage's channel count, and low-pass filtering. Keeping e separate from the scale-specific decoder skips lets the initializer use fused multiscale context while the decoder retains spatially aligned detail. The initializer concatenates c and q, then applies layer normalization, a 3×3 convolution, GELU and a gated block to produce z_0.

Partition-only encoding. For candidate generation, the same encoder weights are applied to each checkerboard view through mask-aware convolutions. If M is the binary validity field and K is a convolution kernel with spatial area q, a masked convolution computes its response from $M \odot x$, rescales it by $q/\max(n, 1)$, where n is the number of valid input sites in that receptive field, adds bias, and zeros outputs whose receptive field contains no valid site. The validity field is propagated by the corresponding test $n > 0$. The local average in the measurement stem is likewise normalized by the number of valid samples. Thus no unobserved pixel value enters a partition encoding, while features may be computed at locations supported by nearby observed samples. During verification, only valid complementary bottleneck features outside the exclusion region contribute to the residual. The full-view encoder and the masked encoder share parameters; the former constructs the deployed base state, while the latter constructs proposal and scoring views.

B.2 Screen-Wall Transport Coupling Learning

In weak wall-observation scenarios, diffuse mixing obscures the origin of screen-dependent variations; meanwhile, direct reconstruction can conflate these variations with static scene appearance. SWTCL therefore organizes the inverse problem around a scene-shared latent transport kernel. To expose the physical motivation, consider a fixed geometry with diffuse wall reflection, direction-independent screen radiance and a linear radiometric response. For each colour channel, an idealized single-wall-bounce model is

$$\bar{y}(u) = b_s(u) + \int_{\Sigma} K_s(u, \zeta)\ell(\zeta)\, dA_\zeta + n(u), \tag{14}$$

$$K_s(u, \zeta) = \int_{\mathcal{W}} Q_s(u, \omega)\frac{\rho_s(\omega)}{\pi} \tag{15}$$

$$\times \frac{\chi_s(\zeta, \omega)\mu(\zeta, \omega)\mu(\omega, \zeta)}{\|\omega - \zeta\|_2^2}\, dA_\omega. \tag{16}$$

Here $\Sigma, \mathcal{W}$ are the screen and relay-wall surfaces, ζ, ω are their spatial coordinates, and u is a camera-image location. The unknown screen radiance ℓ contributes to the ideal linear measurement $\bar{y}$ through the nonnegative kernel K_s; b_s and n denote static illumination and measurement error. The scene index s fixes geometry. Wall reflectance is ρ_s, path visibility is $\chi_s \in \{0, 1\}$, and the nonnegative camera response Q_s includes wall-to-camera transport and pixel integration. The directional cosine is $\mu(a, b) = [\mathbf{n}_a^\top (b - a)/\|b - a\|_2]_+$ for unit surface normal $\mathbf{n}_a$, with $[t]_+ = \max(t, 0)$; dA is surface area measure. This integral explains superposition, attenuation and spatial mixing under the stated assumptions. Equation 14 explains why screen contributions spread, attenuate and superpose on the relay wall. SWTCL learns the corresponding effective relationship in the feature space encoded from the observed RGB image. This connects the physical mixing process to a reusable correspondence between screen-side and measurement-side locations.

For the available observation, local centring separates slowly varying appearance from spatial variation:

$$b(u) = \frac{1}{|\mathcal{N}(u)|}\sum_{v \in \mathcal{N}(u)} y(v), \qquad d(u) = y(u) - b(u), \tag{17}$$

$$(m, \{f_i\}_{i=1}^L) = \mathcal{X}(y, d). \tag{18}$$

Here $y, b, d \in \mathbb{R}^{B \times C_0 \times H_p \times W_p}$ are the padded observation, local average and differential field, where (H_p, W_p) are the padded dimensions. The fixed neighbourhood $\mathcal{N}(u)$ counts reflected boundary samples with multiplicity; degenerate small inputs use $b = y$. The learned representation map $\mathcal{X}$ returns the measurement field $m \in \mathbb{R}^{B \times C \times h \times w}$ and scale evidence $f_i \in \mathbb{R}^{B \times C_i \times H_i \times W_i}$, where L counts scales. The measurement field m and multiscale features $\{f_i\}$

provide the evidence used by routing, proposal generation and reconstruction. Subsequent equations act independently on each sample and suppress the batch index.

Let $N = P^2$ be the number of routing-grid sites and r the number of nonnegative modes. Normalized source and destination factors define an affinity, from which a sparse transport measure is obtained:

$$A_{\alpha\beta} = \sum_{\gamma=1}^{r} U_{\alpha\gamma}\lambda_\gamma V_{\beta\gamma}, \qquad \sum_{\alpha} U_{\alpha\gamma} = \sum_{\beta} V_{\beta\gamma} = 1, \tag{19}$$

$$T_\alpha \in \operatorname*{arg\,max}_{T\subseteq\{1,\dots,N\},\,|T|=k} \sum_{\beta\in T} A_{\alpha\beta}, \tag{20}$$

$$C_{s,\alpha\beta} = g\,\frac{\mathbf{1}[\beta \in T_\alpha]\exp(A_{\alpha\beta}/\tau)}{\sum_{\beta'\in T_\alpha}\exp(A_{\alpha\beta'}/\tau)}. \tag{21}$$

Here $U, V \in \mathbb{R}_+^{N\times r}$ are learned normalized factors, $\lambda \in \mathbb{R}_+^r$ weights the modes, and $g, \tau > 0$ control total row mass and concentration. Factor normalization is written away from numerical denominator floors. Source and destination indices are α, β, while γ indexes modes; T_α retains the k strongest affinities in a row. The derived $C_s \in \mathbb{R}_+^{N\times N}$ satisfies $C_s\mathbf{1} = g\mathbf{1}$ and has at most k nonzeros per row. Only the affinity has rank at most r; exponentiation and support restriction do not preserve that rank. The coupling assigns each screen-side site α a sparse set of measurement-side destinations β and is shared across contents of scene s, so changing the page changes the routed features while retaining the learned spatial correspondence. Parameters are learned during training and frozen across contents of the same scene at inference.

The kernel acts on spatial fields through finite-dimensional quadrature. For a field at latent resolution, define averaging weights and reconstruction basis weights by

$$\mathsf{A}_{\alpha v} = \frac{\mathbf{1}[v \in D_\alpha]}{|D_\alpha|}, \qquad \mathsf{B}_{u\alpha} \geq 0, \quad \sum_{\alpha} \mathsf{B}_{u\alpha} = 1, \tag{22}$$

$$\Phi = \mathsf{B}C_s^\top\mathsf{A}, \qquad \Psi = \mathsf{B}C_s\mathsf{A}, \tag{23}$$

$$(\Phi z)_\nu(u) = \sum_{\alpha,\beta,v} \mathsf{B}_{u\beta}C_{s,\alpha\beta}\mathsf{A}_{\alpha v}z_\nu(v). \tag{24}$$

Here D_α is the averaging neighbourhood associated with routing site α, $\mathsf{A} \in \mathbb{R}^{N\times hw}$ is its fixed analysis matrix, and $\mathsf{B} \in \mathbb{R}^{hw\times N}$ contains bilinear reconstruction weights. Each channel ν is represented as a spatial column vector; Φ transports screen features toward wall coordinates and Ψ transports measurement evidence back toward screen coordinates. Their weights depend on spatial resolution but not on image content. For any routing-grid row vector $v \in \mathbb{R}^N$, nonnegativity gives $\|vC_s\|_1 \leq g\|v\|_1$, bounding redistribution at that grid. This bound does not assert contraction of the complete resampled operator. Moreover, Ψ is generally not the adjoint $\Phi^* = \mathsf{A}^\top C_s\mathsf{B}^\top$ or an inverse. Shared transposed routing therefore expresses parameter consistency, not optical reciprocity.

Using this transport, multiscale evidence and unexplained measurements jointly determine the initial screen state:

$$e = \mathcal{E}\big(\{\Psi(\mathcal{E}_i(f_i))\}_{i=1}^L\big), \qquad c = \Psi(m) + e, \tag{25}$$

$$q = \Psi\big(m - \Phi(c)\big), \qquad z_0 = \mathcal{S}(c, q). \tag{26}$$

Here the learned maps $\mathcal{E}_i$ align channel coordinates, and $\mathcal{E}$ aligns spatial resolutions and aggregates transported evidence. The fields e, c, q, z_0 denote multiscale evidence, routed context, residual context and initialization. The learned map $\mathcal{S}$ takes the channel concatenation of c, q. The residual context q returns the unexplained measurement response to screen coordinates. The initializer combines it with the bottleneck-aligned multiscale context e, while the decoder receives the scale-specific routed features $\{h_i\}$. These pathways connect coarse screen organization with spatial detail during recovery. Table 3 evaluates their reconstruction contributions.

Coupling regularization controls the concentration and spatial variation of the learned routes:

$$\pi_{\alpha\beta} = C_{s,\alpha\beta}/g, \qquad \Gamma_e = -\frac{\sum_{\alpha,\beta}\pi_{\alpha\beta}\log\max(\pi_{\alpha\beta}, \epsilon_e)}{N\log\max(2,k)}, \tag{27}$$

$$\Gamma_v = \sum_{d=1}^{4}\langle|D_d\mathsf{C}|\rangle + \sum_{d=1}^{2}\langle|D_dM|\rangle, \qquad M_\beta = \sum_{\alpha} C_{s,\alpha\beta}. \tag{28}$$

Here π is the row probability, Γ_e its normalized entropy, and $\epsilon_e > 0$ a logarithm floor. The tensor $\mathsf{C} \in \mathbb{R}^{P\times P\times P\times P}$ arranges the coupling along source and destination spatial axes; $M \in \mathbb{R}^{P\times P}$ is destination mass. The fixed D_d takes a nearest-neighbour difference along axis d, and angle brackets average over available differences, with empty averages defined as zero. The scalar Γ_v penalizes variation of both routes and accumulated destination mass. Bounds on mode weights and gain further control the strength of transport. Screen supervision and transport consistency across individual contents, content differences and convex mixtures connect the learned routes to reconstruction. Appendix B.5 gives the training configuration and reproducibility settings. Inference uses one observation and frozen parameters; a new physical geometry requires a new scene coupling.

B.3 Counterfactual Transport Verification and Selective Write-back

In ambiguous reconstruction scenarios, a plausible local detail can have little support in the indirect observation; meanwhile, unconstrained refinement can accumulate unsupported changes. CTVSW therefore treats refinement as a sequence of bounded hypotheses tested against complementary evidence. SWTCL supplies both the full-observation initialization and the partition-conditioned states, while CTVSW returns the accepted state and its spatial support for the final readout. Table 3 evaluates associated image changes, not the physical correctness of every accepted detail.

Let Π_p be complementary checkerboard masks on the latent lattice Λ, with $\Pi_1 = 1 - \Pi_0$, and let $\widehat{\Pi}_p$ be their nearest-site extension to the observed image. Partition-conditioned statistics and representations are

$$b_p(u) = \frac{\sum_{v \in \mathcal{N}_0(u)} \widehat{\Pi}_p(v) y(v)}{\max(1, \sum_{v \in \mathcal{N}_0(u)} \widehat{\Pi}_p(v))}, \tag{29}$$

$$d_p(u) = \widehat{\Pi}_p(u)\big(y(u) - b_p(u)\big), \tag{30}$$

$$(m_p, \{f_{p,i}\}, v_p) = \mathcal{X}_{\Pi_p}(y). \tag{31}$$

Here p identifies the proposal partition, and $\mathcal{N}_0(u)$ includes only in-image neighbours. The shared representation map $\mathcal{X}_{\Pi_p}$ uses only samples from this partition throughout encoding. Normalizing each local aggregation by its available sample count preserves this restriction while accounting for missing observations. The validity field $v_p \in \{0,1\}^{B\times 1\times h\times w}$ identifies features with observed support. The resulting $m_p, f_{p,i}$ have the corresponding full-view feature shapes. Applying the context construction separately yields partition states $z^{(p)}$ and contexts $c^{(p)}$, which generate proposals without consulting the complementary observations. The initial reconstruction z_0 still uses the full observation.

Candidate locality can be characterized geometrically, independently of the parameterization of the proposal maps:

$$u_j^* \in \arg\max_{u \in \Lambda} \varphi_j(u; z^{(p)}, c^{(p)}), \tag{32}$$

$$\Omega_j \in \arg\min_{S \subseteq \Lambda,\, |S| = K} \sum_{u \in S} \|u - u_j^*\|_2^2, \qquad K = \max(1, \lceil \kappa hw \rceil), \tag{33}$$

$$\delta_j \in \{\Delta : \mathrm{supp}(\Delta) \subseteq \Omega_j,\ \|\Delta\|_\infty \le \delta_1\}. \tag{34}$$

Here j indexes precomputed candidates, φ_j is a learned scalar location score, and u_j^* is a selected lattice centre. The set Ω_j contains exactly K nearest sites, with ties resolved within the minimizing set; $\Omega_j(u)$ also denotes its binary indicator. The fraction $\kappa \in (0,1]$ controls support size, and $\delta_1 > 0$ bounds the candidate field δ_j. The learned proposal map specifies candidate amplitudes, and the nearest-site construction places them on a compact support around the selected centre. All candidates are generated from the partition-conditioned states before verification begins. This gives each hypothesis a defined location, support and maximum strength before its measurement effect is tested.

Scoring further excludes a neighbourhood of the proposal partition:

$$\mathcal{P}_p = \{u \in \Lambda : \Pi_p(u) = 1\}, \qquad \mathcal{B}_\infty(R) = \{v : \|v\|_\infty \le R\}, \tag{35}$$

$$V_p(u) = (1 - \Pi_p(u)) v_{1-p}(u) \mathbf{1}[u \notin \mathcal{P}_p \oplus \mathcal{B}_\infty(R)]. \tag{36}$$

Here $R \ge 0$ is a configured lattice radius, $\oplus$ is the Minkowski sum, and $\mathcal{B}_\infty(R)$ is a square neighbourhood. The binary V_p has the same shape as v_p and retains valid complementary sites beyond this exclusion region. Proposal and verification use complementary input samples, while the full observation supplies the base state z_0.

To prevent cumulative drift, each signed hypothesis is projected into an entrywise neighbourhood of the initialization before its effect is evaluated:

$$\mathcal{B}(z_0, \delta_2) = \{\widetilde{z} : \|\widetilde{z} - z_0\|_\infty \le \delta_2\}, \tag{37}$$

$$z_j^{\pm} = \arg\min_{\widetilde{z} \in \mathcal{B}(z_0, \delta_2)} \frac{1}{2} \|\widetilde{z} - (z_{j-1} \pm \delta_j)\|_F^2. \tag{38}$$

Each signed hypothesis is clipped around z_0 in closed form, enforcing the cumulative entrywise amplitude bound before verification.

The discrepancy is the absolute residual integrated over the available verification measure:

$$\mathcal{R}_p(z) = \frac{\sum_{\nu=1}^{C} \sum_{u \in \Lambda} V_p(u) |m_{1-p,\nu}(u) - (\Phi z)_\nu(u)|}{\max(C \sum_{u \in \Lambda} V_p(u), 1)}, \tag{39}$$

$$\rho_{0j} = \mathcal{R}_p(z_{j-1}), \qquad \rho_{\pm j} = \mathcal{R}_p(z_j^{\pm}). \tag{40}$$

Here m_{1-p} is complementary measurement evidence and $\mathcal{R}_p$ is a scalar per sample; the mask is broadcast over channels. The nonnegative $\rho_{0j}, \rho_{+j}, \rho_{-j}$ compare the current state with the actual capped alternatives in the same latent measurement domain. Empty verification support is rejected by the acceptance rule, irrespective of the stabilized denominator.

Sign selection and normalized evidence improvement are then

$$z_j^\star = \begin{cases} z_j^+, & \rho_{+j} \le \rho_{-j}, \\ z_j^-, & \rho_{+j} > \rho_{-j}, \end{cases} \tag{41}$$

$$\xi_j = \frac{\sum_{\nu,u} V_p(u) |[\Phi(z_j^+ - z_{j-1})]_\nu(u)|}{\max(C \sum_u V_p(u), 1)}, \tag{42}$$

$$\eta_j = \frac{\rho_{0j} - \min(\rho_{+j}, \rho_{-j})}{\rho_{0j} + \xi_j + \epsilon}. \tag{43}$$

Here $z_j^\star$ is the lower-residual alternative, $\xi_j \ge 0$ is a reference measurement change, and $\epsilon > 0$ stabilizes the dimensionless improvement η_j. The denominator scales the gain by both the current discrepancy and the transport effect of the bounded positive update. This reference is defined before choosing the update direction, so sign selection compares the two residuals on a common scale. Its asymmetric definition and tie convention are specified in Appendix B.4. All residuals are evaluated in the complementary measurement representation.

We accept a candidate when $\eta_j > \theta$ and it satisfies the admissibility condition $\mathcal{A}_j$. The amplitude and support constraints define the admissible local refinements; Appendix B.4 gives the numerical checks.

The sequential update and accepted-support field satisfy

$$z_j = z_{j-1} + a_j(z_j^\star - z_{j-1}), \qquad \|z_j - z_0\|_\infty \le \delta_2, \tag{44}$$

$$G(u) = \max_{1 \le j \le J} a_j \Omega_j(u) \mathbf{1}[\max_\nu |(z_j - z_{j-1})_\nu(u)| > \epsilon_g], \tag{45}$$

$$a_j = 1 \implies \mathcal{R}_p(z_j) < \mathcal{R}_p(z_{j-1}) - \theta(\rho_{0j} + \xi_j + \epsilon). \tag{46}$$

Here J is the total number of candidates, $G \in \{0,1\}^{B \times 1 \times h \times w}$ is the union of accepted supports with actual change above $\epsilon_g > 0$, and the phase p is the one associated with candidate j. The state bound follows inductively because a hard decision retains the current state or selects a projected state. The residual inequality follows directly from the acceptance threshold, but concerns only that candidate's fixed verification mask; changing partitions does not establish descent of one global objective. An accepted support region does not activate unchanged sites. During training, location and acceptance use straight-through probability surrogates with the same hard forward decisions. For acceptance, the surrogate derivative with respect to η_j is $\sigma_j(1-\sigma_j)/t$, where $\sigma_j = (1 + \exp[-(\eta_j - \theta)/t])^{-1}$ and $t > 0$ is a training temperature. This differentiation device is not an inference-time optimization procedure.

Finally, the spatial support of accepted changes determines where the refined reconstruction contributes to the output:

$$x_0 = \mathcal{D}(z_0, \{h_i\}), \qquad x_1 = \mathcal{D}(z_J, \{h_i\}), \tag{47}$$

$$w(u) = \sum_{v \in \Lambda} \mathsf{B}^x_{uv} G(v), \qquad \mathsf{B}^x_{uv} \ge 0, \quad \sum_v \mathsf{B}^x_{uv} = 1, \tag{48}$$

$$\widehat{x}(u) = (1 - w(u)) x_0(u) + w(u) x_1(u). \tag{49}$$

Here $\mathcal{D}$ is the shared decoder, and $h_i \in \mathbb{R}^{B \times C_i \times H_i \times W_i}$ supplies transported wall evidence at each decoding scale. Both reconstructions use the same evidence, so their difference arises from the verified latent updates. The map B^x interpolates accepted support to image coordinates, giving $w \in [0,1]$. Consequently $w(u) = 0$ implies $\widehat{x}(u) = x_0(u)$, and $|\widehat{x}(u) - x_0(u)| = w(u)|x_1(u) - x_0(u)|$ channelwise. This identity describes how the accepted support controls the spatial contribution of the refined branch. Appendix B.4 specifies resampling and output conventions.

Inference uses a single observation and frozen maps. The accepted latent state determines the refined reconstruction, and its support determines where that reconstruction contributes. Together, they carry complementary-measurement verification through to the final screen image.

B.4 Numerical Implementation

Direction selection and normalization. Equal residuals select the positive update. The reference ξ_j is the transport effect of the actual bounded positive update, including when the negative update has the lower residual. It is computed after imposing the cumulative bound; the uncapped proposal magnitude and the average of the two signed effects would define different scores. Linearity of the frozen Φ gives $\Phi(z_j^+ - z_{j-1}) = \Phi z_j^+ - \Phi z_{j-1}$ in exact arithmetic.

Admissibility and numerical safeguards. The condition $\mathcal{A}_j$ in Section 3 is $u_{1j}u_{2j}u_{3j} = 1$, where

$$u_{1j} = \mathbf{1}[\|\delta_j\|_\infty \le \delta_1 + \epsilon_v]\mathbf{1}\left[\frac{\sum_{\nu,u}(1-\Omega_j(u))|\delta_{j,\nu}(u)|}{Chw} \le \epsilon_v\right], \tag{50}$$

$$u_{2j} = \mathbf{1}[\textstyle\sum_u \Pi_p(u) > 0]\mathbf{1}[\sum_u V_p(u) > 0], \tag{51}$$

$$u_{3j} = \mathbf{1}[\|z_j^\star - z_{j-1}\|_\infty > \epsilon_v], \qquad a_j = \mathbf{1}[\eta_j > \theta]u_{1j}u_{2j}u_{3j}. \tag{52}$$

The test u_{1j} checks the candidate amplitude and spatial support, u_{2j} requires nonempty proposal and verification regions, and u_{3j} excludes updates with negligible actual change. The proposal construction already enforces the amplitude and support constraints; their numerical checks allow tolerance $\epsilon_v > 0$. In the implementation, the amplitude/support tolerance is 10^{-6}, and an actual selected change must exceed 10^{-6} in maximum absolute value. These safeguards preserve the intended feasible set under finite precision. The threshold $\theta \ge 0$ is the required residual improvement. Empty verification regions are rejected even though residual denominators are stabilized. The change threshold $\epsilon_g > 0$ defines which accepted sites enter the output support G.

The relative-improvement score uses $\epsilon = 10^{-4}$ and an acceptance threshold $\theta = 0.02$. The acceptance surrogate uses temperature 0.02, and the add/delete surrogate uses temperature 0.05. Two proposals are generated from each partition, giving four candidates for sequential verification. The surrogate supplies training gradients while preserving the hard forward decisions. Equal residuals select the positive update in the forward pass.

Image reconstruction. At each of four decoding stages, the latent field is upsampled by a 1×1 convolution and pixel shuffle, concatenated with its corresponding routed skip, fused by a 1×1 convolution, and processed by gated blocks. Both branches use the same low-pass-filtered, transported skips and the same decoder and sigmoid output layer. The bilinear map B^x upsamples G and removes input padding to align the gate with the original image extent. The RGB outputs lie in the unit range, with final clipping protecting numerical bounds.

B.5 Training Objective and Settings

The training formulation combines supervision of the base and final reconstructions with transport consistency, candidate supervision, verification and coupling regularization. Transport consistency compares predicted and encoded wall features for individual samples from the same scene, their differences and their convex mixtures. These comparisons encourage the scene coupling to explain changes in screen content through corresponding changes in wall evidence. Screen targets supervise learning, while reconstruction uses the wall-mediated observation and the learned scene coupling.

Training configuration. We train the backbone specified in Appendix A.1 together with the scene coupling and the proposal, verification and readout components. The training settings use AdamW with $\beta = (0.9, 0.99)$, a cosine schedule from 1×10^{-4} to 1×10^{-6}, 200 epochs, batch size 8 and automatic mixed precision, with validation after every epoch. The transport and verification mechanisms use coupling rank 8, a 16×16 routing grid retaining the 32 strongest destinations per row, two candidate proposals per phase, partition block size 4, exclusion radius 1, candidate support fraction 0.25, per-step and cumulative amplitude caps 0.25 and 0.35, and an acceptance margin of 0.02. We use the same architecture and training hyperparameters across the five benchmarks. Each fixed capture geometry is assigned a scene index s and coupling C_s, which is reused across contents acquired in that geometry.

C Extended Experimental Results

The main result raises two further questions: which screen structures carry the reconstruction gain, and how does recovery change when the wall evidence changes? Sections C.1 and C.2 connect additional fidelity measures to visible screen structure, extending the qualitative comparisons with nine Scene A examples and eight Scene B examples. Sections C.3–C.5 examine changes to the captured wall observation through Gaussian noise, exposure scaling and multiplicative speckle. Section C.6 then changes the physical capture conditions and learns the corresponding scene couplings. Together, these results connect recovered content to the wall-mediated evidence and capture conditions that support it.

C.1 Extended Fidelity Metrics

The main comparison reports PSNR, SSIM and LPIPS. Two further fidelity measures are reported here for the same models, split and post-processing: multiscale structural similarity (MS-SSIM) and mean per-image root-mean-square error, $\mathrm{RMSE} = M^{-1}\sum_{i=1}^{M}\sqrt{\mathrm{MSE}_i}$, where M is the number of evaluated images. MS-SSIM examines structure across scales, while RMSE measures image-wise pixel error.

Table 4 lists both metrics for the twenty models, grouping and row order of Table 2. TCCRL attains the best value in all ten columns. On MS-SSIM the margins over the strongest baseline of each column are $+0.0013$ on CtM, $+0.0273$ on SmM, $+0.0627$ on browser, $+0.0276$ on char and $+0.0504$ on software; on RMSE the corresponding reductions are 0.00226, 0.01094, 0.01793, 0.02847 and 0.02070. The narrowest margin is again on CtM, where DRSformer reaches 0.6690 MS-SSIM and 0.13848 RMSE. The largest MS-SSIM gain occurs on browser, at 0.0627 over Restormer, while the largest RMSE reduction occurs on char, at 0.02847. The advantage therefore persists when we examine both structure across scales and image-wise pixel error.

Table 4: Extended fidelity metrics on the five benchmarks: MS-SSIM↑ and RMSE↓. MS-SSIM is computed with the same implementation used throughout; RMSE is averaged after taking the square root of each image's mean squared error. The model set, grouping and row order match Table 2. Red/blue values denote best/second best per column; matching row shading indicates at least one such rank.

Method	CtM		SmM		browser		char		software	
	MS-SSIM↑	RMSE↓	MS-SSIM↑	RMSE↓	MS-SSIM↑	RMSE↓	MS-SSIM↑	RMSE↓	MS-SSIM↑	RMSE↓
Methods for Hidden-Scene Imaging										
Flatland	0.5670	0.15474	0.7472	0.10315	0.7071	0.11829	0.7477	0.23672	0.7180	0.13821
LEAP	0.6378	0.14360	0.8243	0.08236	0.8425	0.07857	0.9255	0.14082	0.8096	0.10569
NLOSNeuS	0.3632	0.22658	0.4573	0.21141	0.4342	0.23821	0.3438	0.33611	0.4366	0.24324
NLOST	0.4622	0.18575	0.4800	0.21704	0.4929	0.19328	0.4537	0.31371	0.5064	0.21267
NLOS-OT	0.6317	0.14449	0.8120	0.08704	0.8497	0.07573	0.8586	0.18484	0.7911	0.11621
Under-scanning	0.4476	0.19020	0.5208	0.19798	0.5141	0.18491	0.7030	0.26502	0.5679	0.19758
Virtual scanning	0.6555	0.13970	0.7750	0.09832	0.5919	0.15932	0.7823	0.22602	0.7050	0.14777
General-Purpose Image Restoration										
AdaIR	0.5314	0.16359	0.7968	0.09449	0.7456	0.11155	0.8017	0.21452	0.7158	0.14094
HAT	0.3689	0.21221	0.3931	0.24127	0.4113	0.24960	0.3280	0.33976	0.4077	0.25512
NAFNet	0.6476	0.14187	0.8218	0.08535	0.8140	0.08876	0.9041	0.15579	0.8022	0.11056
Restormer	0.6419	0.14154	0.8252	0.08384	0.8578	0.07355	0.9362	0.13043	0.8168	0.10261
SFHformer	0.6191	0.14484	0.6807	0.11879	0.6374	0.13658	0.7709	0.23163	0.6973	0.14457
U-Net	0.4462	0.19480	0.4116	0.24173	0.3743	0.25441	0.7979	0.21977	0.5197	0.22655
Uformer	0.5746	0.15677	0.8082	0.08888	0.7859	0.09645	0.8383	0.19722	0.7503	0.12977
VarFormer	0.3876	0.20103	0.4478	0.21677	0.4305	0.23057	0.5873	0.31007	0.4890	0.23072
Models for Related Degradations										
DarkIR	0.6210	0.14659	0.8037	0.08921	0.8245	0.08414	0.8796	0.17201	0.7830	0.11734
DRSformer	0.6690	0.13848	0.8137	0.08691	0.6145	0.15567	0.9062	0.15590	0.7526	0.12860
PhoCoLens	0.5629	0.15759	0.7968	0.09255	0.7655	0.10370	0.8217	0.20163	0.7356	0.13354
VolFormer	0.3506	0.25527	0.3551	0.27646	0.3212	0.32072	0.3272	0.33866	0.3577	0.30533
Ours (TCCRL)	0.6703	0.13622	0.8525	0.07142	0.9205	0.05562	0.9638	0.10196	0.8672	0.08191

C.2 Extended Qualitative Comparison

Figure 3 spans a single example per dataset, which is enough to read the failure modes but not enough to show that they repeat. Figures 11 and 12 extend the same two comparisons to nine and eight further examples, one sample per row, with the same column order, the same border convention and the same per-example PSNR readout as Figure 3. Four examples are shown for *SmM*, *char*, *browser* and *software*, and one for *CtM*. The examples were selected by hand to expose the failure modes described below, not drawn at random, so they should be read as illustrations rather than as an unbiased sample.

Scene A links reconstruction gains to the separation of screen structures. TCCRL has the highest displayed PSNR in all nine examples. The character cases show clearer separation of several strokes and glyphs, with margins of 2.1–4.4 dB. The SmM cases retain the arrangement of windows, tables and large colour regions, with margins of 2.5–3.8 dB. The CtM example preserves the arrangement of concentric coloured bands and reaches 15.09 dB, compared with 14.83 dB for the strongest displayed baseline. Across these contents, the wall response supports both regional organization and localized shape.

Scene B changes where the evidence is visible while retaining the hidden-screen task. The input contains the display housing and the surrounding wall response. In Figure 12, TCCRL preserves the arrangement of browser panels and chart elements, together with the flower outline and broad photographic structure in the software examples. These reconstructions connect the visible wall margins to content located on the unseen emitting surface.

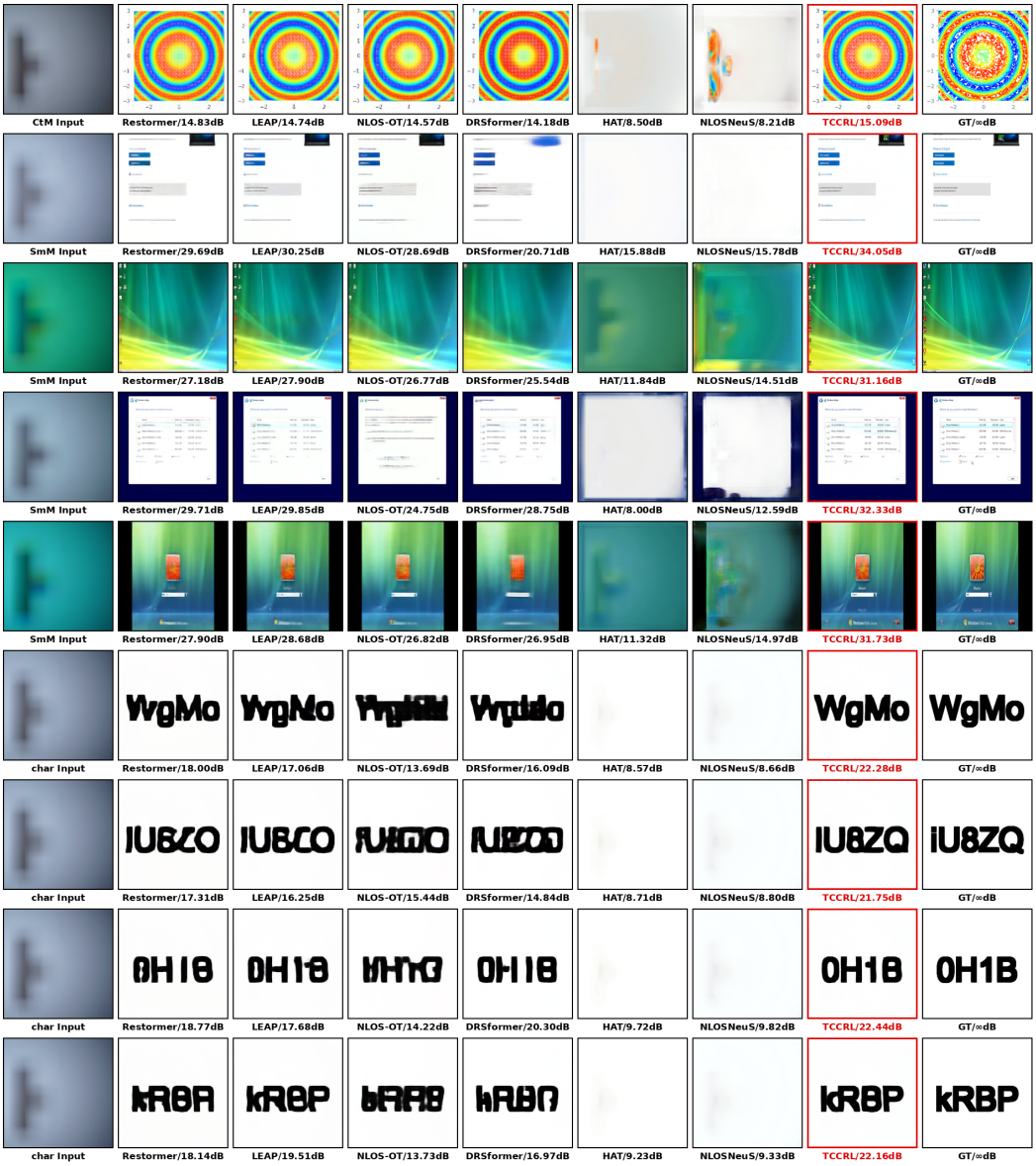


Figure 11: Extended qualitative comparison for the Scene A arrangement of Figure 3. The nine rows are one sample from *CtM* and four samples each from *SmM* and *char*, one sample per row, with the dataset name carried in the input-cell label of each row exactly as in Figure 3. Columns follow Figure 3: input, six baselines, TCCRL (red border) and ground truth. Values below the reconstructions are the per-example PSNR in dB, recomputed for this figure against the ground-truth panel, so they need not reproduce the corpus-level averages of Table 2; ∞ under GT denotes self-comparison.



Figure 12: Extended qualitative comparison for the Scene B arrangement of Figure 4. The eight rows are four samples each from *browser* and *software*, matching the row order of Figure 4. Columns are those of Figure 4 rather than Figure 11: input, the three baselines displayed there, TCCRL (red border) and ground truth. These samples are 512×288, so the cells are laid out at 16:9 as in Figure 4; border convention and PSNR readout follow Figure 11.

C.3 Robustness to Additive Gaussian Noise

Figures 14–18 give the Gaussian-noise curves for all five metrics and all five benchmarks over the plotted range. The sweep spans $\sigma = 0$ to 0.03 at ten levels. Figures 14–18 show the five levels $\{0, 0.003, 0.006, 0.010, 0.013\}$; the remaining levels are summarized below. The $\sigma = 0$ level reproduces the clean operating point of Table 2. Figures 12–16 show TCCRL and the 14 baselines identified in the legend. The full-set rank summaries additionally include Flatland, Under-scanning, HAT, U-Net and NLOST.

Figure 13 connects small changes in the wall observation to changes in recovered character structure. At the strongest displayed level, input PSNR remains about 37.7 dB, while reconstruction PSNR changes by more than 10 dB. The paired views make the weak content-bearing response visible through its effect on the reconstruction.

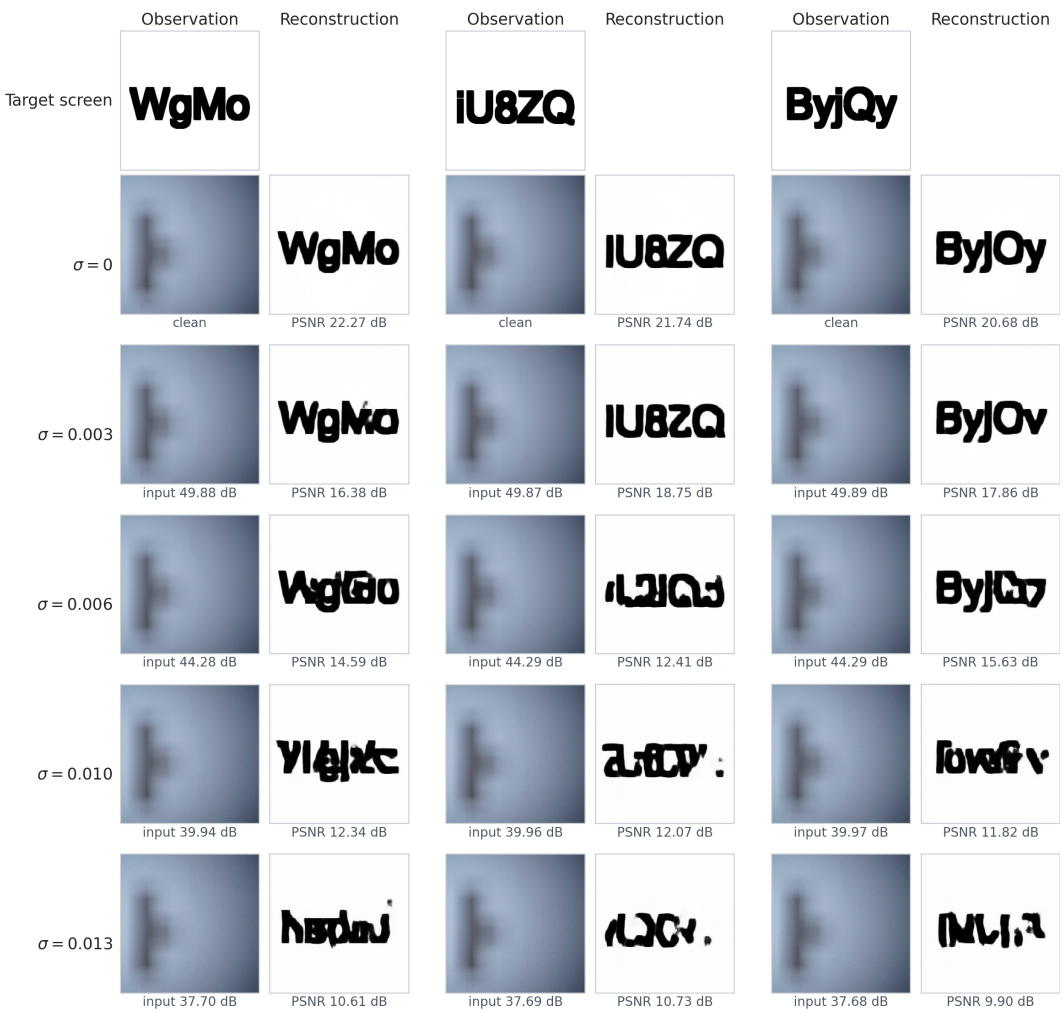


Figure 13: Gaussian-noise robustness, qualitative. Rows correspond to the five displayed Gaussian-noise levels. Three char examples are shown, each with an observation–reconstruction pair. The target screen is shown once at the top of each example. Input PSNR compares perturbed and clean observations; reconstruction PSNR compares recovered and target screen images. At the strongest displayed level, input PSNR is about 37.7 dB while reconstruction PSNR changes by more than 10 dB.

The noise sweep shows that page and mixed-content reconstruction retain a substantial advantage as the wall observation is perturbed. Across the plotted levels, TCCRL is first throughout the five metrics on SmM, browser and software. At $\sigma = 0.013$, TCCRL retains PSNR margins of 1.62 dB on SmM, 2.89 dB on software and 1.01 dB on browser. CtM and char show the largest rank changes at the stronger displayed level; the complete per-level values remain in the curves and tables.

Across the sweep, the changing reconstructions connect recovered screen structure to the weak content-dependent response carried by the wall observation. Read together with the clean-image comparisons, these results characterize how evidence quality shapes hidden-screen recovery.

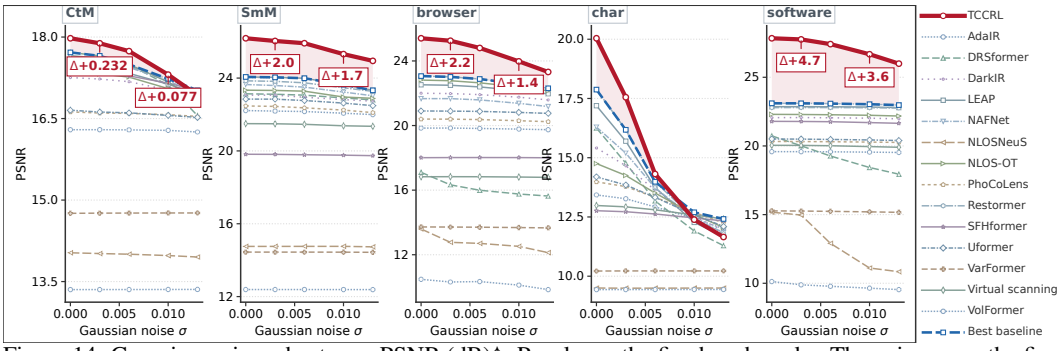


Figure 14: Gaussian-noise robustness, PSNR (dB)↑. Panels are the five benchmarks. The axis covers the five milder levels of the sweep, $\sigma \leq 0.013$. Red: TCCRL; dashed blue: the best of the 14 plotted baselines at each level; shading and the Δ labels mark the levels where TCCRL is ahead. All curves use the fifteen-model pool of the main-text robustness figure.

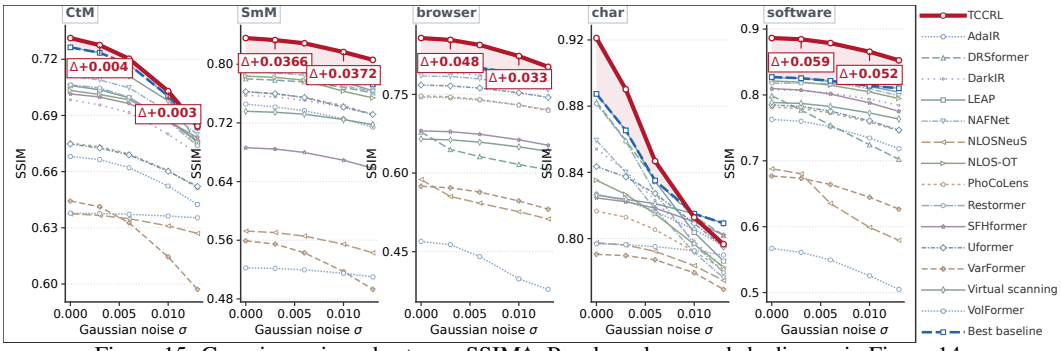


Figure 15: Gaussian-noise robustness, SSIM↑. Panels, colours and shading as in Figure 14.

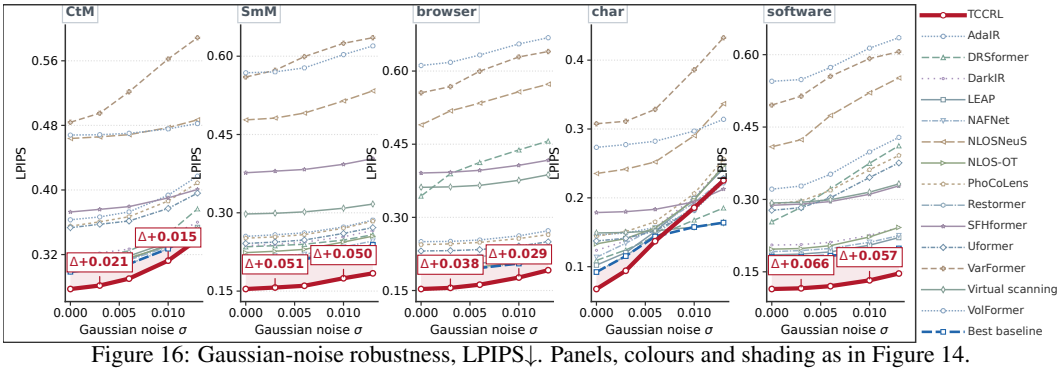


Figure 16: Gaussian-noise robustness, LPIPS↓. Panels, colours and shading as in Figure 14.

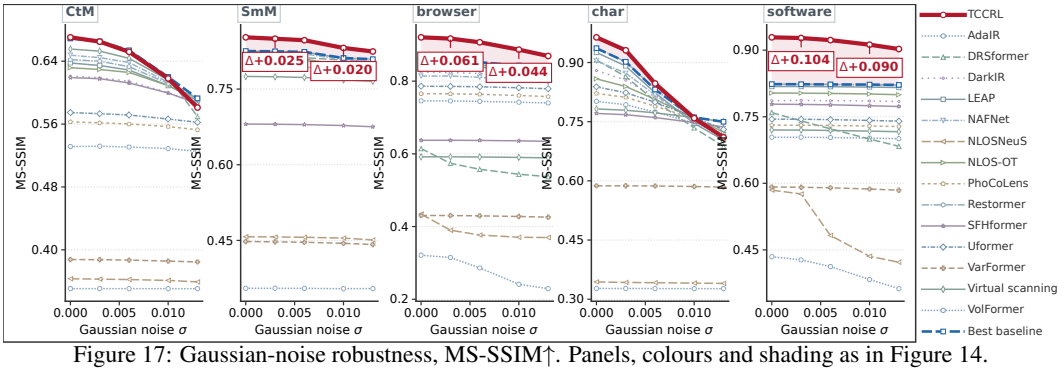


Figure 17: Gaussian-noise robustness, MS-SSIM↑. Panels, colours and shading as in Figure 14.

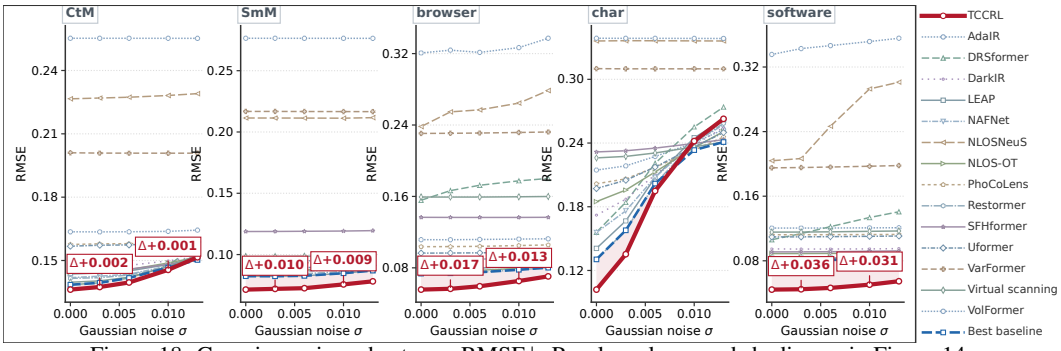


Figure 18: Gaussian-noise robustness, RMSE↓. Panels, colours and shading as in Figure 14.

C.4 Robustness to Exposure Shift

The exposure sweep changes the intensity scale of the captured wall observation. We multiply the observation by 2^{EV}, with $\mathrm{EV} \in \{-0.5, -0.375, \ldots, +0.5\}$. Figures 20–24 show the inner five levels ($|\mathrm{EV}| \leq 0.25$); the remaining levels are summarized below. This isolates the response to a global intensity change.

Figure 19 shows the same sweep qualitatively. The rows are ordered from over-exposure to under-exposure with the clean operating point in the middle, so the reconstruction readout peaks in the middle row: the two arms are not symmetric, and the figure is drawn on *software*, the benchmark the sweep costs most in PSNR.

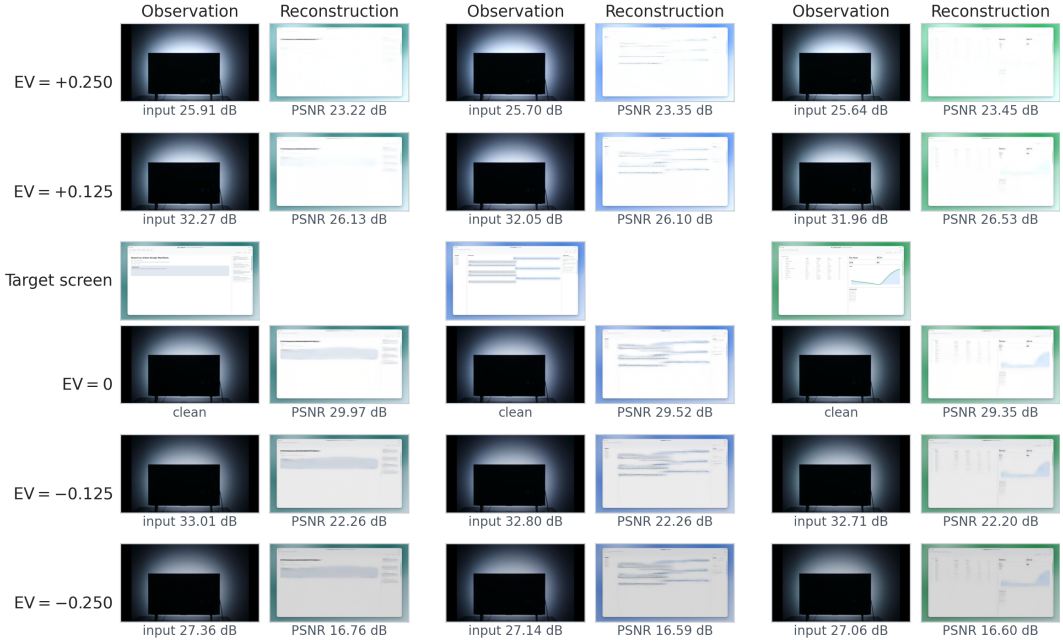


Figure 19: Exposure-shift robustness, qualitative. The target screen is shown once at the top of each example. Rows then run from $\mathrm{EV} = +0.25$ to $\mathrm{EV} = -0.25$, with the clean observation at $\mathrm{EV} = 0$. Each example uses an observation–reconstruction pair; Input PSNR compares perturbed and clean observations, and reconstruction PSNR compares the reconstruction with the target screen.

Across the 125 benchmark–metric–level comparisons, TCCRL ranks first in 112 comparisons against the full baseline set and in 120 against the plotted baseline set. browser retains first place across all five metrics and all five displayed exposure levels. Over-exposure is nearly free: at $\mathrm{EV} = +0.25$ TCCRL is first on every metric of every benchmark except PSNR, MS-SSIM and RMSE on CtM, where it is second. Under-exposure is where the pixel-wise rankings vary; the complete values are shown in the curves.

Exposure scaling separates sensitivity to intensity from retention of spatial structure. Its effect varies across screen contents, while the structural-similarity results remain strong across the evaluated gains. CtM begins with the narrowest clean PSNR margin in Table 2, and its ordering changes with the imposed gain. On software, the PSNR reductions are 7.74 dB under under-exposure and 4.51 dB under over-exposure; on char, they are 5.10 dB and 0.56 dB, respectively.

The outer four levels are reported here rather than drawn. At $\mathrm{EV} = -0.5$ the structural metrics still hold, while the pixel-wise rankings vary by benchmark; at $\mathrm{EV} = +0.5$ no metric falls below second except LPIPS on char, which is third. The exposure sweep therefore distinguishes changes in overall intensity from added spatial variation in the measurement.

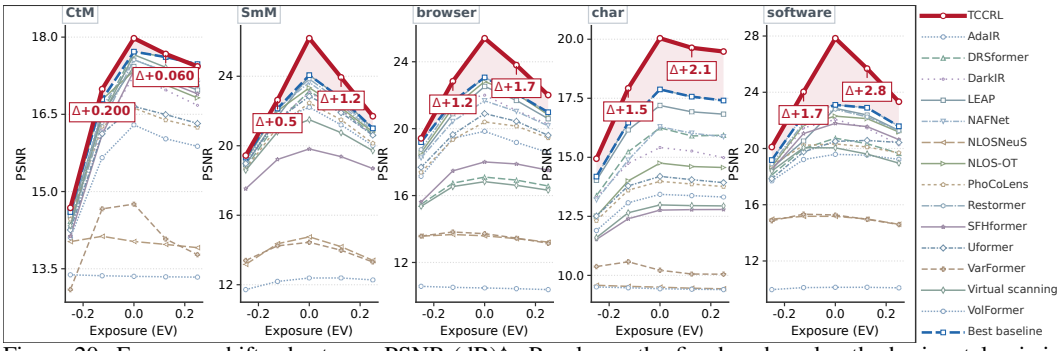


Figure 20: Exposure-shift robustness, PSNR (dB)↑. Panels are the five benchmarks; the horizontal axis is the exposure gain in EV, covering the inner five levels of the sweep, $|\mathrm{EV}| \leq 0.25$. Curves and shading as in Figure 14.

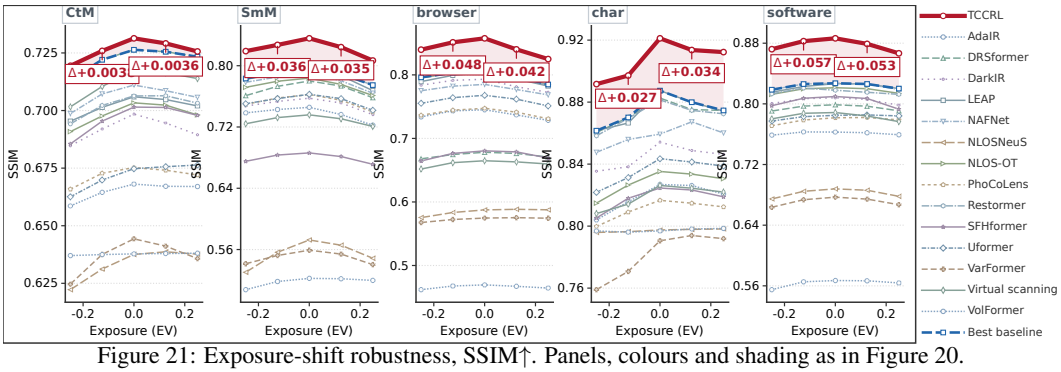


Figure 21: Exposure-shift robustness, SSIM↑. Panels, colours and shading as in Figure 20.

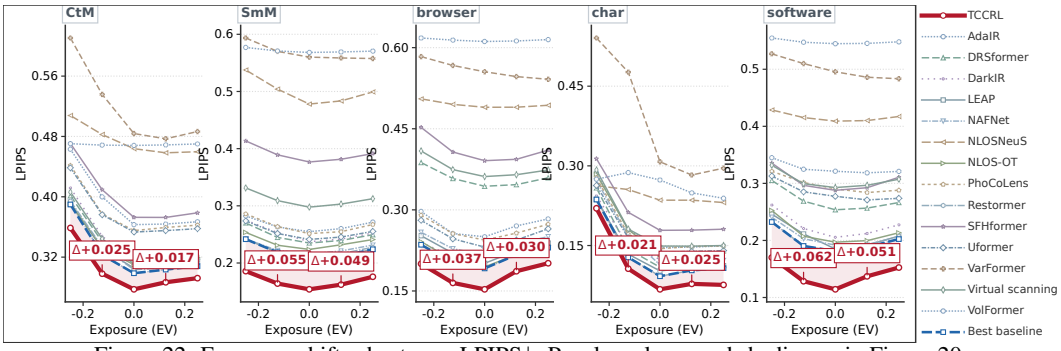


Figure 22: Exposure-shift robustness, LPIPS↓. Panels, colours and shading as in Figure 20.

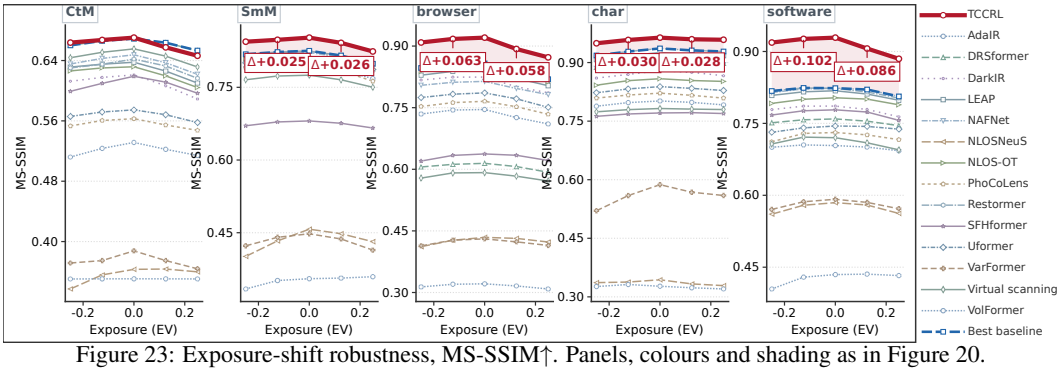


Figure 23: Exposure-shift robustness, MS-SSIM↑. Panels, colours and shading as in Figure 20.

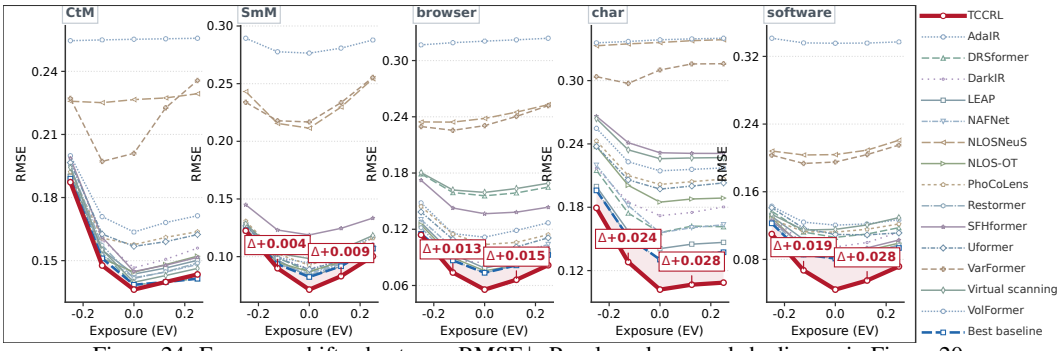


Figure 24: Exposure-shift robustness, RMSE↓. Panels, colours and shading as in Figure 20.

C.5 Robustness to Multiplicative Speckle

This subsection examines signal-dependent spatial fluctuations through four multiplicative-speckle severities at a fixed coherence length of 2 px. Input PSNR expresses the perturbation strength on a common scale.

Figure 25 pairs each perturbed SmM observation with its reconstruction, connecting the visible speckle to the recovery of window layouts and image structure.

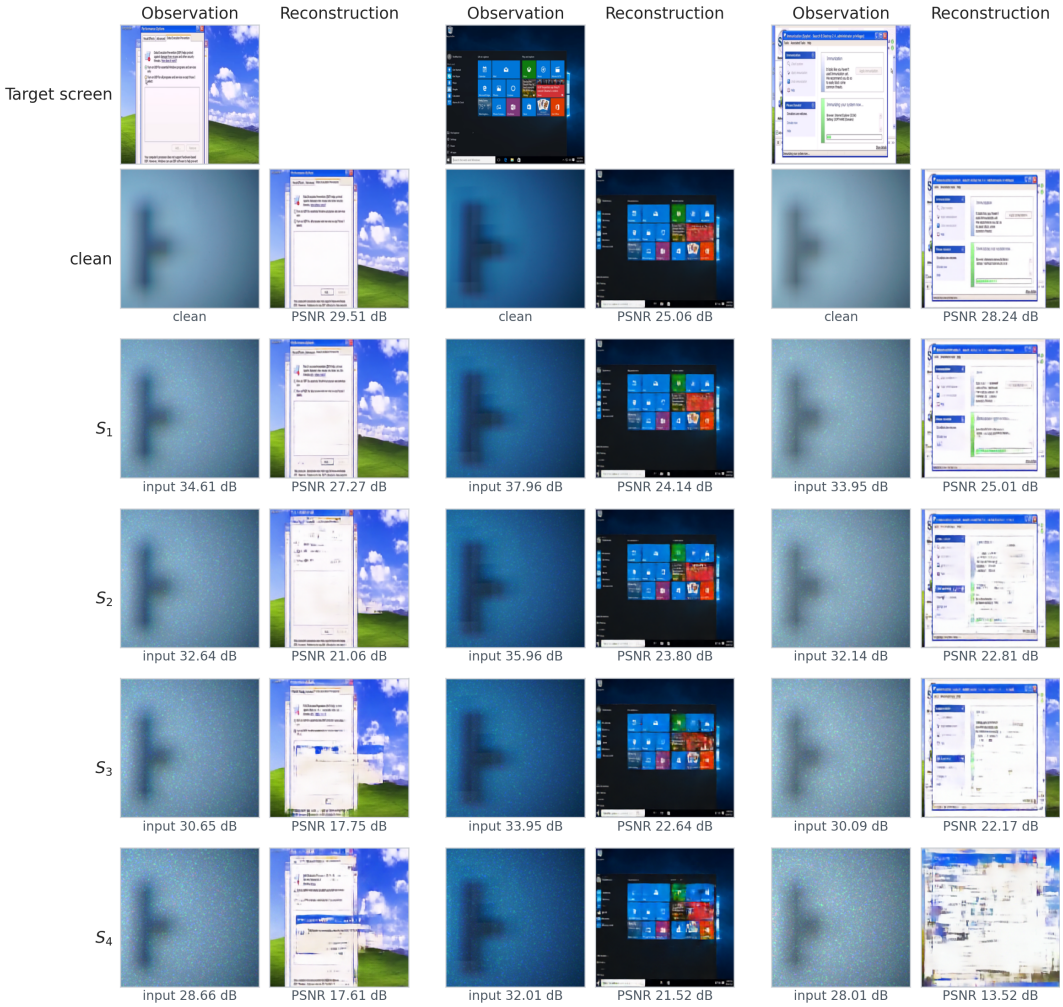


Figure 25: Multiplicative-speckle robustness on SmM, qualitative. Rows are the clean observation and the four speckle severities S_1–S_4, and columns are the ones of Figure 13 — two per sample, *Observation* and *Reconstruction* — for three displayed samples, with each sample's *Target screen* directly above its clean observation at the top of the figure. The two readouts are also the ones of Figure 13, so the clean row's observation columns are marked *clean*. Input PSNR is computed separately for each displayed example against its clean observation using the corresponding 8-bit images.

C.6 Robustness to Physical Capture Geometry

The three sweeps above perturb the captured observation arithmetically. This subsection reports the complementary study, in which the physical condition of the capture is changed and the scene is re-acquired: the distance between the camera and the relay wall, the angle between the optical axis and the wall normal, and the reflectance of the wall surface. Each condition uses its own scene ID and is trained and evaluated with its own scene coupling, so a point on these curves is a new physical scene rather than a degraded version of an existing capture. Figures 26–34 report the three axes, one figure per axis and metric, over all five benchmarks.

On the distance axis the camera is moved along the wall normal, keeping orientation, focus, screen and wall fixed, at $d_{cw}/d_0 \in \{0.85, 0.925, 1.00, 1.075, 1.15\}$; the reference level is the interior point 1.00, so this axis has two arms rather than a monotone sequence. Figure 26 shows the PSNR curves, Figure 27 the SSIM curves and Figure 28 the LPIPS curves. The angle axis tilts the camera about the same wall observation point, holding focal length and exposure, at 0° to 30° off the normal in steps of 5° up to 20° and a final step to 30°, with 0° as the reference (Figures 29–31). The material axis replaces the wall surface at $\rho_d \approx 0.70$ (reference), 0.60 and 0.50, and adds a fourth condition at the nominal reflectance of the reference surface with a fixed fine texture applied to it (Figures 32–34); the fourth condition is a separate material class rather than the next point of the reflectance series, so the horizontal axis of those three figures is ordinal, with the measured reflectance printed as the tick label.

The outcome of all three axes is consistent across the evaluated conditions. Over the 19 baselines of Table 2, TCCRL attains the best value at every level of every axis, on all five benchmarks and in all three metrics, giving 45 curves and 225 compared readings in total. Across the whole study the smallest margin over the strongest baseline is on CtM, where the clean column is already the narrowest: $+0.0050$ and $+0.2604$ at the reference level of the distance axis, in SSIM and PSNR respectively. The largest PSNR margin is on software: $+4.59$ dB at the reference distance and $+4.86$ dB at the largest tilt. The shaded envelope uses the 14 plotted baselines; its best-baseline value is at or below the full-set best for PSNR and SSIM, and at or above it for LPIPS.

On the angle axis, the PSNR margin on char widens from $+2.17$ dB at 0° to $+2.44$ dB at 30°, and the SSIM margin on SmM widens from $+0.0374$ to $+0.0464$. On the distance axis, the char SSIM margin is $+0.0337$ at $1.00d_0$, compared with $+0.0405$ at $0.85d_0$ and $+0.0410$ at $1.15d_0$. On the material axis, the char LPIPS margin grows from $+0.0247$ to $+0.0301$. The physical sweeps extend the reconstruction result across different capture conditions. Each condition has its own scene coupling, allowing the evidence route to reflect that geometry and wall surface. On software, PSNR stays between 27.77 and 27.84 dB across the distance sweep, a range of 0.07 dB. Together with the angle and material comparisons, this shows that the route-and-verify approach can establish hidden-screen recovery across the evaluated physical scenes.

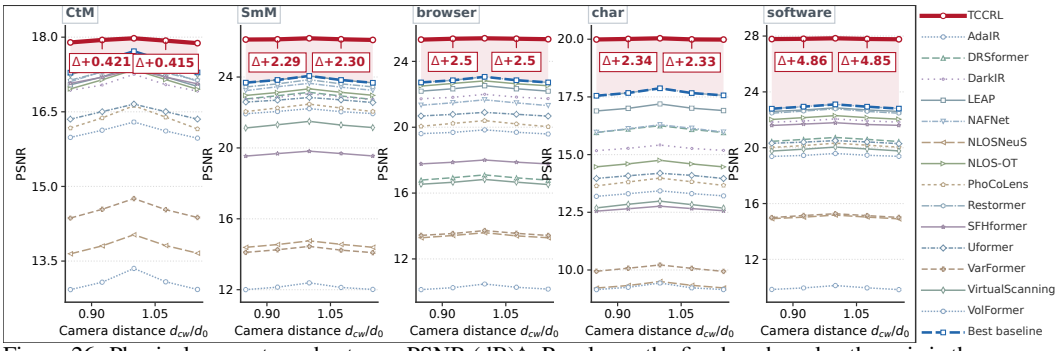


Figure 26: Physical-geometry robustness, PSNR (dB)↑. Panels are the five benchmarks; the axis is the camera-to-wall distance d_{cw}/d_0, whose reference level is the interior point 1.00. Red: TCCRL; dashed blue: the best of the 14 plotted baselines at each level; shading and the Δ labels mark the levels where TCCRL is ahead, as in Figure 14. The plotted models are those of Figure 7.

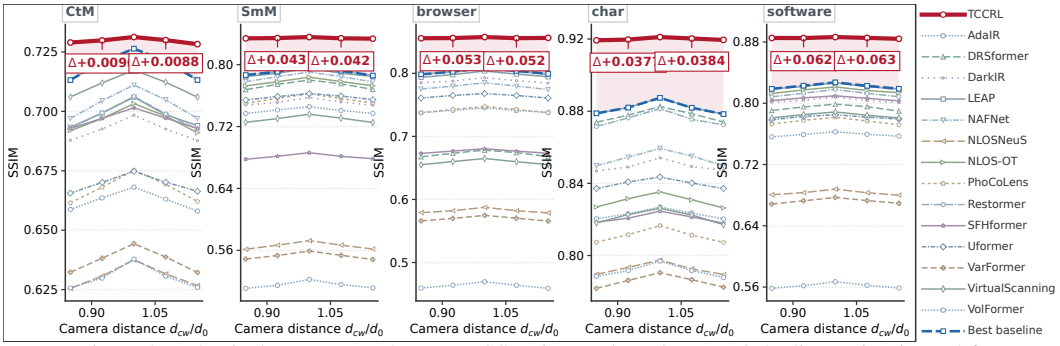


Figure 27: Physical-geometry robustness, SSIM↑. Panels, colours and shading as in Figure 26.

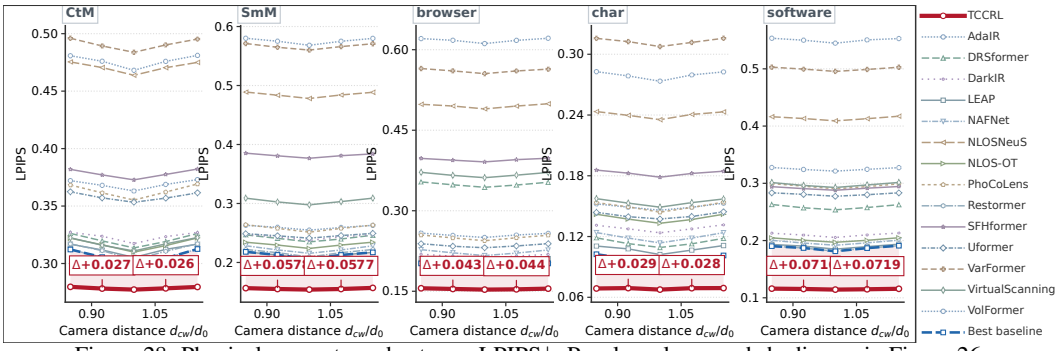


Figure 28: Physical-geometry robustness, LPIPS↓. Panels, colours and shading as in Figure 26.

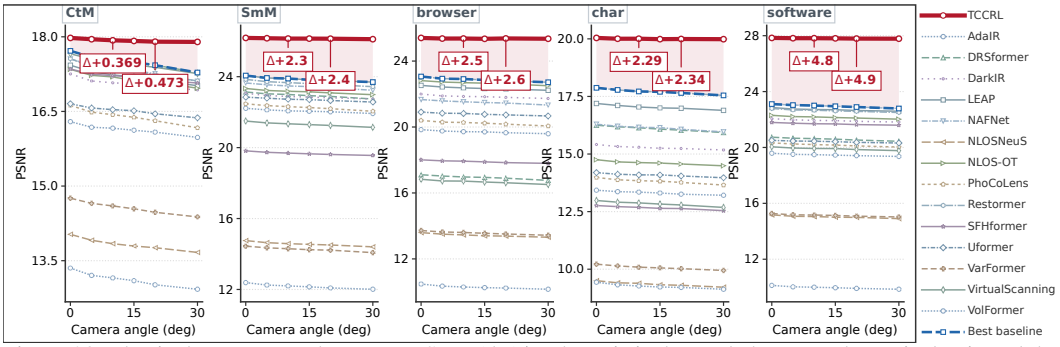


Figure 29: Physical-geometry robustness, PSNR (dB)↑. The axis is the angle between the optical axis and the wall normal, in degrees, with 0° as the reference. Panels, colours and shading as in Figure 26.

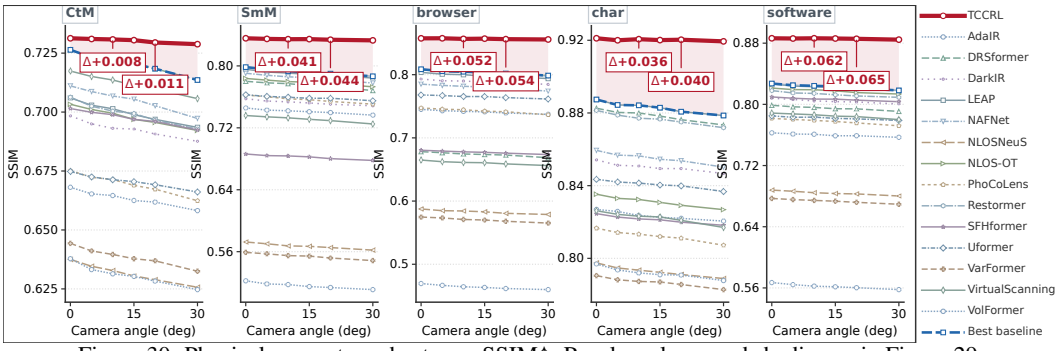


Figure 30: Physical-geometry robustness, SSIM↑. Panels, colours and shading as in Figure 29.

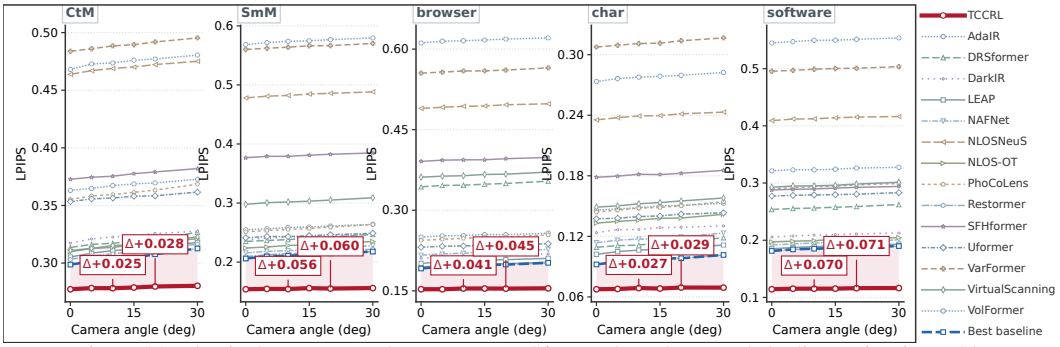


Figure 31: Physical-geometry robustness, LPIPS↓. Panels, colours and shading as in Figure 29.

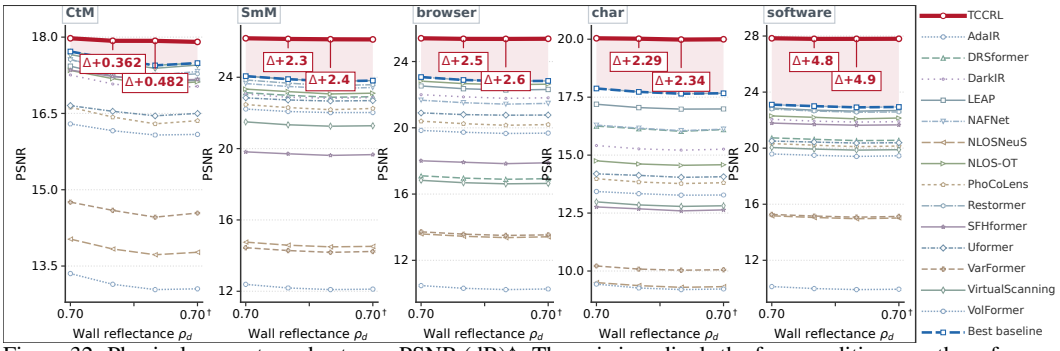


Figure 32: Physical-geometry robustness, PSNR (dB)↑. The axis is ordinal: the four conditions are the reference surface at $\rho_d \approx 0.70$, then $\rho_d \approx 0.60$ and 0.50, then the reference reflectance with a fixed fine texture added (†). Panels, colours and shading as in Figure 26.

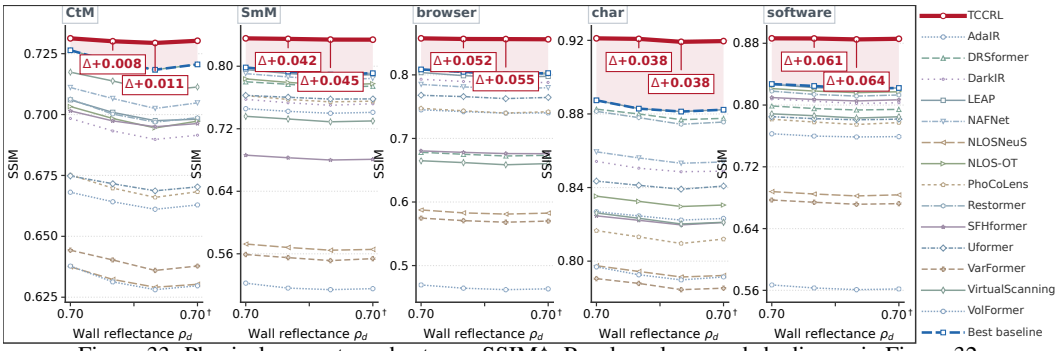


Figure 33: Physical-geometry robustness, SSIM↑. Panels, colours and shading as in Figure 32.

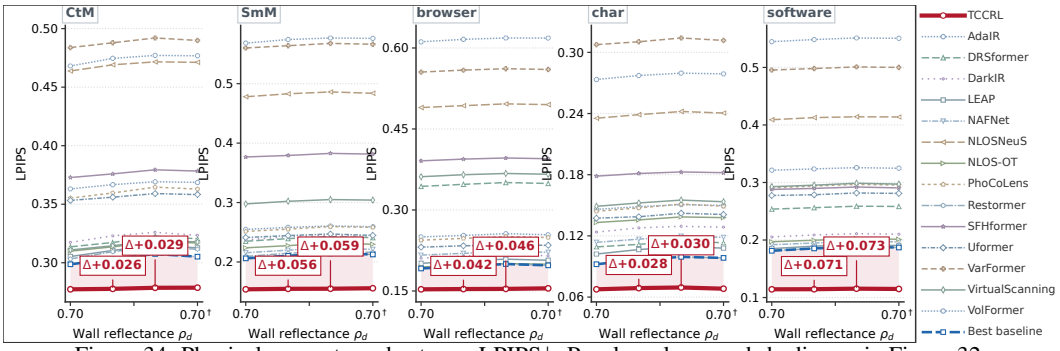


Figure 34: Physical-geometry robustness, LPIPS↓. Panels, colours and shading as in Figure 32.

C.7 Spatial Attribution of the Wall Evidence

The sweeps above perturb the whole observation at once. This subsection asks the complementary question of the trained model: which part of the observation does the reconstruction actually read? We partition the observation into a 3×3 grid with boundaries at $\mathrm{round}(k \cdot s/3)$ for $k = 0, 1, 2, 3$ — which is $[0, 85, 171, 256]$ on both axes of the 256×256 benchmarks and $[0, 96, 192, 288]$ by $[0, 171, 341, 512]$ on the 512×288 ones — and re-run the forward pass with exactly one cell perturbed and the other eight left untouched. Cells are numbered row-major, 1–9. The inference path contains no stochastic sampling, so masking the same cell always yields the same output and the differences measured here are not forward-pass jitter. Every quantity below is a per-sample difference against the unmasked baseline, averaged over the split.

Three fill conditions are used. *Black* sets the cell to zero and *noise* replaces it with i.i.d. $\mathcal{N}(0.5, 0.25)$. Both are destructive and measure robustness rather than attribution: the perturbing strength of the first scales with how bright the cell was to begin with, and the second destroys the local structure outright, so on the two dark Scene-B benchmarks the cell-to-cell spread collapses — the coefficient of variation of the per-cell ΔPSNR falls from 0.387 and 0.437 under black fill to 0.046 and 0.042 under noise fill. The third condition is the attribution condition. *Mean* replaces the cell with that cell's own mean over the split, computed from the val *inputs* only. This removes exactly the part of the cell that varies with the screen content and leaves its static background statistics in place, which is the operational form of the question this paper is about; black and noise also destroy the static part, and so answer a different one.

Masking a ninth of the observation costs something for any model, so each cell is compared against a matched random control: ten repeats of a region whose size is copied from one of the nine cells — so that the area distribution is that of a cell exactly, not merely the same in expectation — placed uniformly at random over the observation, and filled the same way. The control's spread across repeats is narrow (on *CtM* it covers $[+2.70, +2.85]$ dB), and a cell is called separated

below only when it exceeds the control's *largest* repeat. One caveat applies to this control and is worth stating: a uniform top-left placement weights the centre of the frame more heavily than its border, because more placements cover a central pixel than a corner one. Where the centre is the most important region that inflates the control mean and makes most cells look small beside it, which is why the criterion is the strict one and why the structural claims below do not rest on the control alone.

Under mean fill the Scene-B result separates cleanly, and it is the spatial form of the routing claim of Section 3. The cells exceeding all ten control repeats are 2, 4, 6 and 8 on *browser* ($+10.35$, $+10.41$, $+11.97$ and $+13.70$ dB) and 2, 6 and 8 on *software* ($+12.28$, $+10.68$ and $+14.11$ dB). These are the wall bordering the display: the glow above it, the narrow strips to its left and right, and the region below it. Cell 5, whose observation is almost entirely the display's housing rather than the wall, falls below *every* control repeat in both benchmarks — $+6.36$ dB against $[+8.92, +9.47]$ on *browser* and $+5.88$ dB against $[+7.50, +8.16]$ on *software* — and is the cheapest of the middle column on both.

A second reading of the same numbers makes the point without appealing to the control at all. Cell 5 is not a cell whose content stands still: its cross-sample variance is 0.0111 in both benchmarks, the second largest of the nine behind only the 0.032 of cell 2. Masking it nevertheless costs less than masking cells 2 or 8, which bracket it. Cell 8 carries less content variation than cells 2, 5 or 6, yet masking it costs the most of the nine. Normalising the mean-fill ΔPSNR by the cell's own variance — so that what is read is degradation per unit of removed content variation, and cells that simply vary more are not favoured — cell 5 falls to 0.57 and 0.53 on the two benchmarks, second smallest of the nine in both, while cell 8 rises to 2.25 and 1.89, the largest. The housing's observation varies with the screen and the reconstruction does not read it; the wall's varies and it does. Table 6 lists the ratios, and Figure 35 shows the raw ΔPSNR of Figure 5 normalised per dataset and averaged within each arrangement, so that the three Scene-A and the two Scene-B benchmarks carry equal weight and the two panels can be read side by side. In that pooled view the four dearest cells of the Scene-B panel are exactly the four that border the display — cells 8, 2, 6 and 4, in that order — and cell 5 is below all four of them.

Under the Scene-A arrangement the attribution is concentrated rather than diffuse, but the experiment cannot resolve the same distinction there. On *CtM* the dearest cell costs $+4.54$ dB and the cheapest $+0.53$ dB, a factor of 8.6; on *char* the factor is 98, from $+7.83$ down to $+0.08$. This concentration is not a consequence of where the observation happens to vary: the nine cells' cross-sample variances lie within a factor of 1.26 on *CtM* and 1.68 on *char*, so the input carries content-dependent variation almost evenly across the frame while the reconstruction's dependence on it is not even. As on Scene B, only cells 4 and 5 exceed all ten control repeats, and on *CtM* and *char* the remaining seven fall below the control's smallest repeat. What the arrangement does not permit is the housing-versus-wall reading: here the camera images the device's shadow on the wall, and there is no separately visible emitting surface and housing surface whose contributions could be told apart. The result therefore supports the weaker statement, that the dependence is confined to the parts of the wall observation carrying the screen-induced response.

Two limitations close the subsection. *SmM* shows the ordering of the other Scene-A benchmarks — cells 4 and 5 dearest, at $+15.36$ and $+16.10$ dB against a control of $[+12.85, +13.91]$ — but with far less contrast: its normalised ratios span only 0.27 to 0.51, so on that benchmark the dependence is broad rather than localised, consistent with content dominated by low-frequency structure that the mask leaves recoverable elsewhere in the frame. And the control is a same-size axis-aligned rectangle, so it bounds the effect of area without matching the effect of shape; a cell and a random placement of equal area need not be equally removable for a model whose routing carries its own spatial prior.

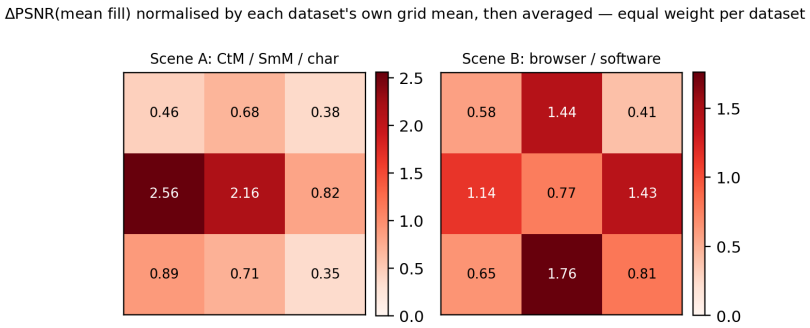


Figure 35: The same attribution, normalised so that the three Scene-A and the two Scene-B benchmarks carry equal weight. Each benchmark's nine ΔPSNR values are divided by its own nine-cell mean before the group average, because the benchmarks' absolute degradations differ by up to a factor of five and an unweighted average of decibels would be dominated by the largest. Under the Scene-A arrangement cells 4 and 5 are the two dearest of every Scene-A benchmark. Under the Scene-B arrangement the four dearest are exactly the cells bordering the display — 8, 2, 6 and 4 at 1.76, 1.44, 1.43 and 1.14 — while cell 5, whose observation is the display's own housing, reaches only 0.77. This figure reads relative importance within a group, not absolute degradation.

Table 5: Spatial attribution of the wall evidence under mean fill. Each entry is the per-sample ΔPSNR (dB) obtained by replacing one ninth of the observation with that cell's own cross-sample mean — the content-dependent part of the cell is removed while its static background statistics are kept — averaged over the split. The last two columns are the matched random control: ten repeats of a region whose size is copied from one of the nine cells and whose location is uniform over the observation, filled the same way. Values in **bold** exceed the control's largest of the ten repeats, so the cell is separated from every random placement of the same size rather than only from the control mean. N is the number of held-out samples.

	Masked cell (row-major, 1–9)									Same-size random control	
Dataset	1	2	3	4	5	6	7	8	9	mean	[min, max]
CtM (N=350)	+0.79	+1.30	+0.55	**+4.54**	**+3.36**	+0.71	+2.51	+1.35	+0.53	+2.77	[+2.70, +2.85]
SmM (N=124)	+10.09	+13.27	+8.06	**+15.36**	**+16.10**	+11.13	+9.53	+12.56	+7.46	+13.54	[+12.85, +13.91]
char (N=453)	+0.08	+0.30	+0.23	**+7.83**	**+6.57**	+2.26	+0.84	+0.55	+0.21	+4.13	[+3.87, +4.27]
browser (N=453)	+4.92	**+10.35**	+3.14	**+10.41**	+6.36	**+11.97**	+6.96	**+13.70**	+7.73	+9.21	[+8.92, +9.47]
software (N=453)	+4.33	**+12.28**	+3.33	+7.77	+5.88	**+10.68**	+3.56	**+14.11**	+5.20	+7.82	[+7.50, +8.16]

Table 6: Same experiment, attribution normalised by how much the mask removed. Cells differ in how strongly their content varies across the split, so the raw ΔPSNR of Table 5 mixes "how much was removed" into "how much the model relied on it". Each entry divides the mean-fill ΔPSNR by that cell's cross-sample variance ($\times 10^3$, so the units are dB per unit of removed content variance); larger means the model depends more on the cell's screen-dependent response. The ratio is comparable across cells within a dataset but not across datasets, whose cell variances span two orders of magnitude (char 2.4×10^{-5}, browser 7.9×10^{-3}). The underlined entry of each row is its largest.

	Masked cell (row-major, 1–9)									Cell cross-sample variance	
Dataset	1	2	3	4	5	6	7	8	9	min	max
CtM	0.29	0.49	0.24	<u>1.81</u>	1.26	0.29	1.01	0.52	0.24	0.0022	0.0028
SmM	0.27	0.39	0.35	<u>0.51</u>	0.45	0.43	0.30	0.38	0.34	0.0221	0.0368
char	3.00	12.05	9.43	193.99	<u>239.82</u>	94.57	23.45	21.66	8.90	2.4×10^{-5}	4.0×10^{-5}
browser	0.62	0.32	0.69	1.39	0.57	1.23	1.88	<u>2.25</u>	1.43	0.0037	0.0319
software	0.59	0.39	0.69	1.13	0.53	1.00	1.01	<u>1.89</u>	0.86	0.0035	0.0318