GSAioIR: Exploring Gaussian Fields for All-in-One Image Restoration

Anonymous authors
Paper under double-blind review

Abstract

Gaussian primitives have advanced 3D scene representation and have shown promise for image-plane modelling. We investigate whether explicit Gaussian primitives can serve as a shared output interface for image restoration and downstream prediction. Existing formulations address task-specific objectives, motivating all-in-one restoration as a setting for testing this interface across degradation operators. Two challenges arise. At fixed primitive density, larger images require more Gaussians, motivating crop-wise prediction with limited global context. Additive rendering depends on accumulated weight, whereas normalised blending with nonnegative weights confines a single RGB field to a convex average of its colours. We introduce GSAioIR, a task-conditioned Gaussian restoration framework. A bounded full-image view and crop-box geometry provide aligned context for local primitive prediction. Three parallel fields produce coarse RGB and signed residuals at the middle and fine scales. Independent normalisation removes uniform weight-scale dependence. Residual summation then permits corrections outside the coarse field's colour convex hull, while content-guided weighting (CGW) uses query–key matching to modulate fine-scale aggregation. Experiments support restoration across five degradation types. Downstream transfer extends the backbone to segmentation, detection, depth, and surface-normal prediction. These results support 2D Gaussian outputs for restoration and downstream vision.

1 Introduction

Explicit Gaussian primitives support 3D scene representation and real-time novel-view synthesis from posed images (Kerbl et al., 2023). Their geometry and appearance are optimised jointly through differentiable rendering. We investigate whether this explicit output representation can also support image restoration and downstream prediction. Image-plane Gaussian models already address representation, compression, and super-resolution (Zhang et al., 2024; Hu et al., 2025; Chen et al., 2025). Restoration-specific studies cover denoising, low light, and dehazing (Jang & Jin, 2026; Chen et al., 2026a;b). Our focus is a shared, task-conditioned Gaussian interface for noise, rain, haze, blur, and low-light restoration, with extensions to segmentation, detection, depth, and surface normals.

Figure 1 summarises three questions that guide this study.

1. Problem transfer. What new challenges emerge when adapting Gaussian representations from 3D reconstruction to 2D restoration? Restoration must infer a clean image from a degraded observation rather than render a scene from a new viewpoint. At fixed primitive density, the number of Gaussians grows with image area. Crop-wise prediction limits the active allocation but must recover context beyond the crop. Image-plane rendering must also combine overlapping Gaussians without a prescribed front-to-back order. Summing their weighted colours makes pixel brightness depend on the total weight. Normalisation removes this dependence, but each pixel remains a weighted average of the primitive colours. We use independently normalised RGB and signed residual fields to allow corrections beyond the coarse field's colour range.

2. Restoration effectiveness. Can a shared Gaussian output interface restore images effectively across different degradation types? We introduce GSAioIR to investigate this question. A bounded full-image view conditions local primitive prediction, and independently normalised RGB and residual fields form the restored image. We evaluate the interface on Task3 and Task5, covering noise, rain, haze, blur, and low illumination. GSAioIR attains higher mean PSNR than the tested GaussianSR

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.

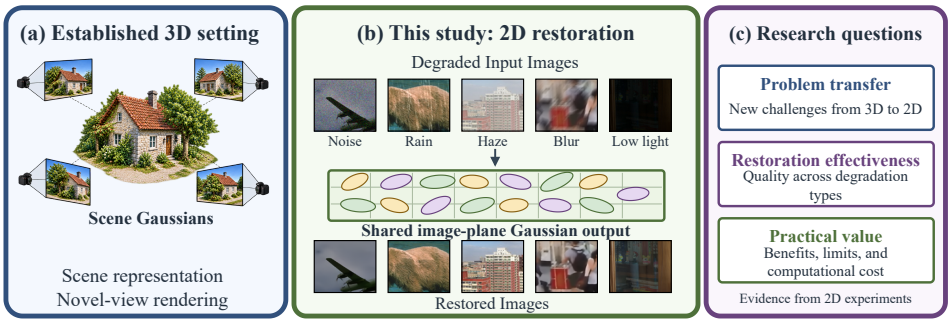

Figure 1: **Research scope of Gaussian outputs.** (a) Established 3D Gaussian scene representation from multi-view observations. (b) Shared image-plane Gaussian outputs for five restoration tasks. (c) Research questions on problem transfer, restoration effectiveness, and practical value.

and GSASR restoration adaptations, denoted GaussianSR-R and GSASR-R. Their training budgets, capacities, and conditioning differ, so this comparison concerns the tested systems rather than an isolated architectural effect. We also compare with dense-output models such as AirNet, PromptIR, InstructIR, and DFPIR, which predict restored RGB images without an explicit Gaussian output field. The main tables retain the original evaluation settings of the reported results. An additional comparison using identical degraded inputs is provided in Appendix B.2. GSAioIR's recorded results lie within the performance range of these models on Task3 and Task5. Deblurring is a relative strength, with the highest recorded GoPro PSNR among the listed Task5 methods (Tables 1 and 2).

3. Practical value. What benefits and computational trade-offs do Gaussian representations offer compared with dense restoration models? Explicit fields allow us to examine how their contributions vary across images through fixed-weight interventions. Profiling reveals lower peak allocated memory but higher latency for the tested tiled implementation relative to full-image DFPIR (Appendix C.2). We further adapt the restoration backbone with task-specific branches for segmentation, detection, depth, and surface-normal prediction. This tests whether its learned features remain useful when the output changes from restored RGB to semantic or geometric predictions.

Our contributions are threefold:

1. **Shared Gaussian outputs.** We investigate a task-conditioned Gaussian interface for restoration across five degradation types, using shared primitive prediction and rendering to connect explicit image representations with all-in-one restoration.
2. **Restoration formulation.** We combine global-to-local context, independently normalised RGB and signed residual fields, and content-guided weighting (CGW) to control output range and adjust contributions among overlapping primitives.
3. **Empirical evaluation.** We compare Gaussian adaptations and dense-output models, analyse field behaviour and computational cost, and evaluate transfer to four downstream tasks. Retrained and fixed-weight controls examine component effects and learned field allocation.

2 Related Work

2.1 Gaussian image representations

Gaussian image models differ in their primitive attributes and conversion to pixels. GaussianImage directly accumulates RGB contributions for representation and compression (Zhang et al., 2024); GaussianSR interpolates latent features before decoding (Hu et al., 2025). GSASR predicts RGB Gaussians from an input image for arbitrary-scale reconstruction (Chen et al., 2025). These image-plane models also differ from visibility-ordered splatting for three-dimensional scenes (Kerbl et al., 2023). Our model renders RGB and signed residual fields with independent normalisation.

Gaussian restoration is not limited to clean-image super-resolution. GRAPE evaluates denoising (Jang & Jin, 2026), and Pixel-to-Gaussian studies joint deraining and super-resolution (Peng et al., 2026). LL-GaussianMap predicts enhancement gain maps (Chen et al., 2026a), while Dehaze-GaussianImage

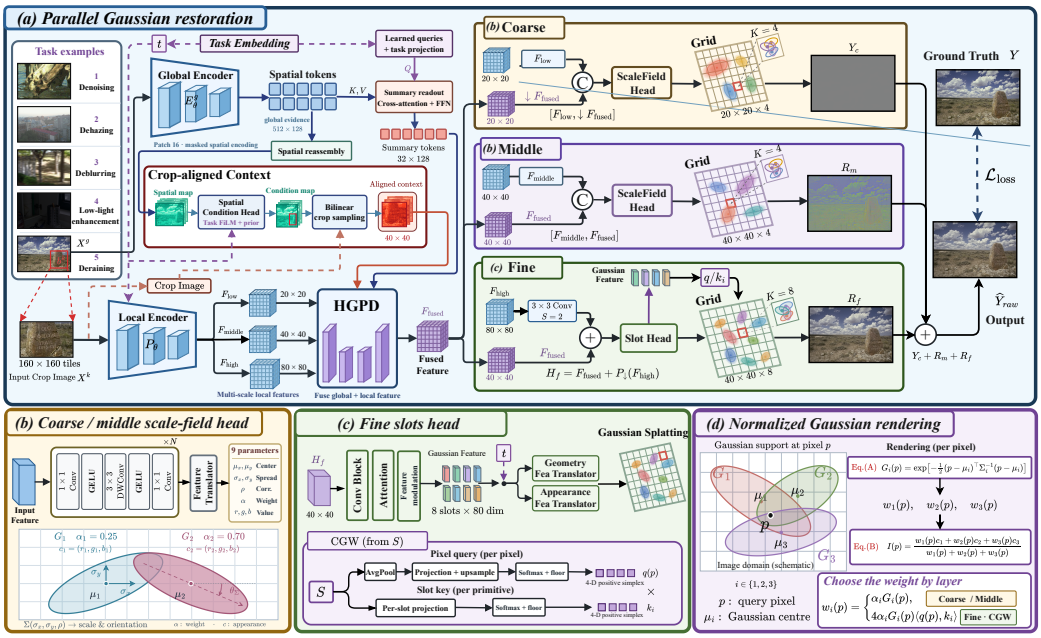

Figure 2: **GSAioIR architecture (Base model).** (a) Global/local encoders and HGPD condition three parallel Gaussian fields. (b) Coarse/middle parameter readout. (c) Fine slots and content-guided weighting (CGW). (d) Gaussian kernel, Eq. (A), and normalised rendering, Eq. (B). The three rendered fields sum to the raw prediction $\widehat{Y}_{\text{raw}}$.

specialises the representation to dehazing (Chen et al., 2026b). Our shared model combines bounded scene context with independently normalised RGB and residual fields.

2.2 All-in-one restoration

Shared restoration models adapt their features through learned corruption representations in AirNet (Li et al., 2022), prompts in PromptIR (Potlapalli et al., 2023), frequency processing in AdaIR (Cui et al., 2025), and expert routing in MoCE-IR (Zamfir et al., 2025). DFPIR applies degradation-guided channel and attention perturbations (Tian et al., 2025). These methods vary feature conditioning; GSAioIR instead studies a shared restoration model with an explicit Gaussian output.

Conditioning information remains a separate comparison axis. InstructIR uses natural-language instructions to guide restoration (Conde et al., 2024). PromptIR performs blind restoration without a supplied corruption label (Potlapalli et al., 2023). GSAioIR receives a task label at training and inference, while the image and crop geometry determine its primitive parameters. We state task information, model scope, and preprocessing for each comparison.

3 Method

GSAioIR predicts 2D Gaussian primitives from a degraded image and a task label. As shown in Figure 2, local features and crop-aligned global context condition three parallel heads. Their coarse RGB and signed residual fields are rendered independently and summed to reconstruct the image.

3.1 Task-conditioned global-to-local features

Let X be a degraded image, Y its clean target, and t the supplied task label. Its learned embedding supplies the condition t shown in Figure 2(a). To bound the active primitive allocation, we process local crops X^k with normalised boxes b^k and a bounded full-image view X^g. The global and local encoders extract scene context and multiscale crop features, respectively:

$$Z^g = E^g_\theta(X^g, t), \qquad (F^k_{\text{high}}, F^k_{\text{middle}}, F^k_{\text{low}}) = P_\theta(X^k, t). \tag{1}$$

The local encoder P_θ in Figure 2(a) uses a Restormer-style architecture (Zamir et al., 2022) and produces 80×80, 40×40, and 20×20 features for a 160×160 crop. In particular, F_{middle} denotes the middle-resolution encoder feature used by HGPD and the middle head. We omit the crop index k from local features below.

The global encoder produces masked spatial tokens and 32 summary tokens. A spatial condition head and bilinear crop sampling align scene features to each crop. For $b^k = (x_0, y_0, x_1, y_1)$, the geometry encoding records the crop centre and extent:

$$g(b^k) = (x_0 + x_1 - 1,\ y_0 + y_1 - 1,\ \log(x_1 - x_0),\ \log(y_1 - y_0)). \tag{2}$$

Fourier features of this geometry augment the task condition, while a position-aware token readout supplements the aligned spatial context. The global branch supplies context; reconstructed RGB comes entirely from the local Gaussian fields, without a degraded-image RGB skip.

3.2 Hierarchical Gaussian Physical Decoder

The Hierarchical Gaussian Physical Decoder (HGPD) fuses the three local feature scales with scene context before Gaussian parameter prediction. It projects F_{high}, F_{middle}, and F_{low} to a common channel width and aligns them to the middle resolution. Their mean initialises the decoder state. Each block mixes the aligned sources with learned weights and incorporates the crop-aligned spatial condition. Task-conditioned feature-wise affine modulation (FiLM) adapts this state to the restoration task. Cross-attention to the global summary tokens supplies scene-level context, and a gated convolutional feed-forward block mixes local spatial features. These residual updates produce the latent feature F_{fused}, which has 320 channels on a 40×40 grid for a 160×160 crop. The Base model configuration and decoder details are given in Appendices A.1 and A.1.

3.3 Parallel 2D Gaussian fields

The coarse and middle heads concatenate encoder and HGPD features; the fine head adds a projected high-resolution skip. Their inputs are

$$H_c = [F_{\text{low}}, \text{pool}(F_{\text{fused}})], \quad H_m = [F_{\text{middle}}, F_{\text{fused}}], \quad H_f = F_{\text{fused}} + P_{\downarrow}(F_{\text{high}}). \tag{3}$$

The pooling operator is the $\downarrow F_{\text{fused}}$ path in Figure 2(a), reducing the grid to 20×20. $P_{\downarrow}$ is a stride-two 3×3 convolution. All heads predict from features rather than a previous rendered image. The coarse/middle ScaleField heads in Figure 2(b) use pointwise and depthwise convolutions followed by a parameter translator. The fine Slot Head in Figure 2(c) converts the latent image features into slot features for Gaussian prediction. Convolution aggregates information across neighbouring grid locations, while slot attention exchanges information among candidate primitives at each location. Feature modulation combines these slot representations with the input feature. Separate task-conditioned geometry and appearance translators then map the abstract features to parameters with explicit Gaussian roles. The geometry translator predicts centres, axis scales, and correlation; the appearance translator predicts positive amplitudes and signed RGB residuals. Together, these predictors connect the learned feature space to the Gaussian representation used by the renderer.

Each primitive has a centre $\mu_i \in \mathbb{R}^2$, covariance $\Sigma_i \in \mathbb{R}^{2\times2}$, positive amplitude α_i, and three-channel appearance c_i. Positive axis scales and a correlation magnitude below one make Σ_i positive definite. These quantities form the nine scalars listed in Figure 2(b). The anisotropic 2D Gaussian kernel is given by Eq. (A) in Figure 2(d), where p and μ_i use the same image-plane coordinates. Its covariance determines spatial support and orientation, while α_i controls its aggregation weight. Coarse appearances are bounded RGB values; middle and fine appearances are signed residuals. Coarse and middle heads predict four primitives per cell on 20×20 and 40×40 grids. Fine prediction shares the 40×40 grid and uses eight 80-dimensional slots in the Base model. These grids provide different supports, not enforced frequency bands. Covariance construction, parameter mappings, and capacity details appear in Appendices A.1 and A.1.

3.4 Content-guided weighting

Geometry-only weights $\alpha_i G_i(p)$ can mix overlapping primitives with different appearances. Content-guided weighting (CGW) adjusts fine-field contributions using compatibility between pixel queries

Table 1: **Task3 restoration.** PSNR (dB)/SSIM. The σ columns refer to CBSD68; Avg. equally averages all five settings.

Method	Venue	SOTS	Rain100L	$\sigma = 15$	$\sigma = 25$	$\sigma = 50$	Avg.
Dense-output references: published results							
FDGAN	AAAI'20	24.71/0.924	29.89/0.933	30.25/0.910	28.81/0.868	26.43/0.776	28.02/0.883
AirNet	CVPR'22	27.94/0.962	34.90/0.967	33.92/0.933	31.26/0.888	28.00/0.797	31.20/0.910
Restormer	CVPR'22	30.43/0.975	36.55/0.975	33.84/0.931	31.18/0.885	27.90/0.790	31.98/0.911
PromptIR	NeurIPS'23	30.58/0.974	36.37/0.972	33.98/0.933	31.31/0.888	28.06/0.799	32.06/0.913
InstructIR	ECCV'24	30.22/0.959	37.98/0.978	34.15/0.933	31.52/0.890	28.30/0.804	32.43/0.913
DFPIR	CVPR'25	31.87/0.980	38.65/0.982	34.14/0.935	31.47/0.893	28.25/0.806	32.88/0.919
Gaussian restoration implementations							
GaussianSR-R	AAAI'25†	25.50/0.919	31.37/0.946	33.36/0.924	30.65/0.871	27.26/0.760	29.6289/0.8842
GSASR-R (9M)	ICCV'25†	27.29/0.945	30.79/0.920	31.80/0.906	29.97/0.859	27.20/0.765	29.4099/0.8792
GSAioIR (ours)	Ours	29.80/0.958	37.35/0.979	34.06/0.935	31.44/0.893	28.22/0.807	**32.1757/0.9143**

Table 2: **Task5 restoration.** Avg. equally averages five tasks; CBSD68 uses $\sigma = 25$. Other conventions follow Table 1.

Method	Venue	SOTS	Rain100L	CBSD68	GoPro	LOLv1	Avg.
Dense-output references: published results							
Restormer	CVPR'22	24.09/0.927	34.81/0.962	31.49/0.884	27.22/0.829	20.41/0.806	27.60/0.881
TAPE	ECCV'22	22.16/0.861	29.67/0.904	30.18/0.855	24.47/0.763	18.97/0.621	25.09/0.801
IDR	CVPR'23	25.24/0.943	35.63/0.965	31.60/0.887	27.87/0.846	21.34/0.826	28.34/0.893
AirNet	CVPR'22	21.04/0.884	32.98/0.951	30.91/0.882	24.35/0.781	18.18/0.735	25.49/0.846
InstructIR	ECCV'24	27.10/0.956	36.84/0.973	31.40/0.887	29.40/0.886	23.00/0.836	29.55/0.907
DFPIR	CVPR'25	31.64/0.979	37.62/0.978	31.29/0.889	28.82/0.873	23.82/0.843	30.64/0.913
Gaussian restoration implementations							
GaussianSR-R	AAAI'25†	24.08/0.892	30.47/0.935	30.41/0.864	25.49/0.778	18.56/0.740	25.8023/0.8418
GSASR-R (9M)	ICCV'25†	25.95/0.925	27.15/0.850	28.73/0.832	25.30/0.772	18.84/0.749	25.1957/0.8259
GSAioIR (ours)	Ours	29.73/0.958	36.65/0.976	31.23/0.888	29.48/0.886	23.02/0.850	**30.0232/0.9113**

and primitive keys, as shown in Figure 2(c). Both derive from the shared slot features S, coupling local image content to the choice of contributing primitives.

Let S_{uvj} denote slot j at grid location (u, v) and $\overline{S}_{uv} = K_f^{-1} \sum_{j=1}^{K_f} S_{uvj}$ its slot average. Learned affine projections W_q, W_k generate query and key logits. We map them to positive four-component vectors as

$$\begin{aligned} \pi_\varepsilon(z) &= (1-\varepsilon)\,\mathrm{softmax}(z) + \frac{\varepsilon}{4}\mathbf{1}, \quad \varepsilon = 0.01, \\ q(p) &= \pi_\varepsilon\big(\mathrm{up}[W_q\overline{S}](p)\big), \qquad k_i = \pi_\varepsilon(W_k S_i). \end{aligned} \tag{4}$$

Here S_i is the slot associated with primitive i, and up bilinearly upsamples query logits to the pixel grid before softmax. The uniform floor keeps every component positive; both vectors sum to one. Their inner product modulates the spatial weight:

$$w_i^f(p) = 4\alpha_i^f G_i^f(p)\langle q(p), k_i\rangle. \tag{5}$$

Thus overlapping primitives can receive different relative weights at the same pixel according to feature compatibility. A uniform query recovers ordinary Gaussian weighting because $4\langle q(p), k_i\rangle = 1$. CGW changes aggregation weights; the appearance translators still predict the signed residual values. Coarse and middle fields retain $w_i^s(p) = \alpha_i^s G_i^s(p)$ for $s \in \{c, m\}$.

3.5 Independently normalised Gaussian rendering

Additive accumulation couples output magnitude to total Gaussian weight. We normalise each field independently, extending the schematic three-primitive average in Figure 2(d), Eq. (B), to all retained

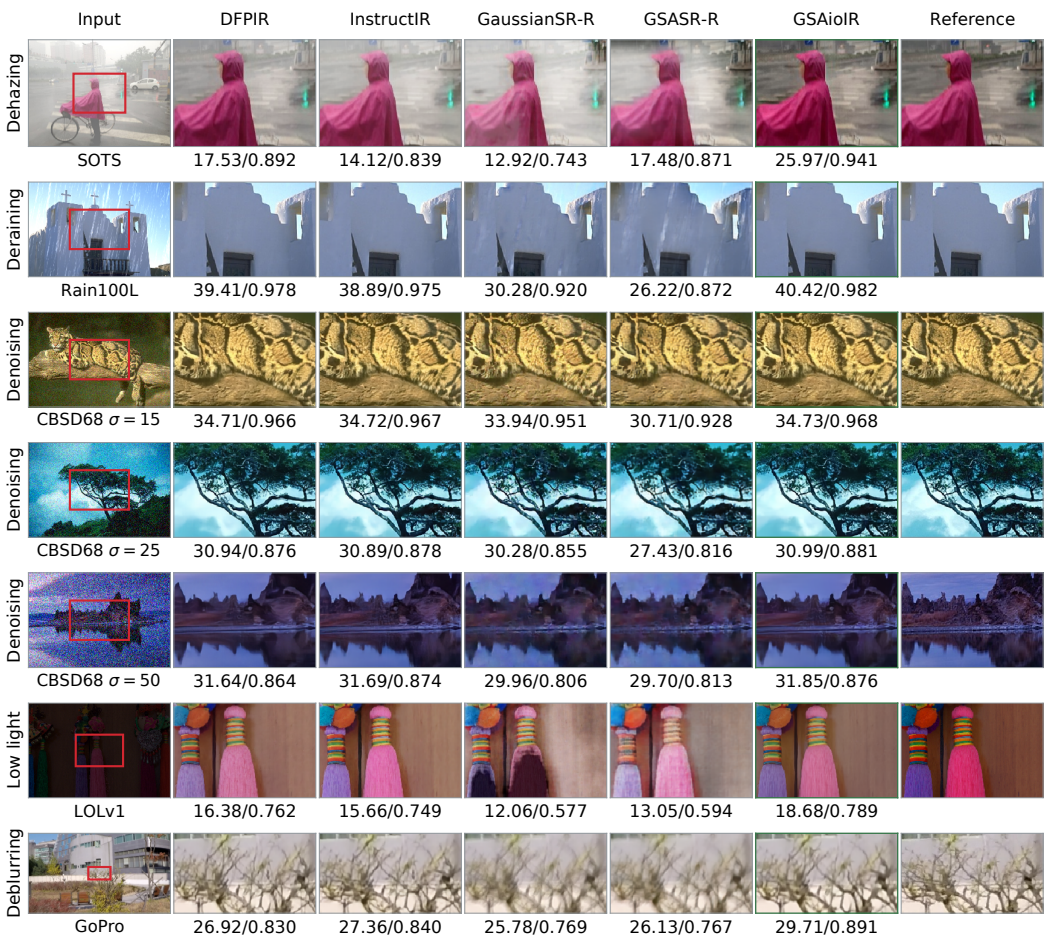

Figure 3: **Restoration examples across five tasks.** Denoising shows different images at $\sigma = 15, 25, 50$. Red boxes mark magnified regions; numbers are full-image PSNR/SSIM.

primitives. The resulting coarse RGB and signed residual fields are

$$Y_c(p) = \frac{\sum_i w_i^c(p) c_i^c}{\sum_i w_i^c(p)}, \quad R_m(p) = \frac{\sum_i w_i^m(p) c_i^m}{\sum_i w_i^m(p)}, \quad R_f(p) = \frac{\sum_i w_i^f(p) c_i^f}{\sum_i w_i^f(p)}. \tag{6}$$

Kernel values at most 10^{-6} are discarded before amplitude weighting. Each sum uses this retained support; a zero denominator returns zero. On fixed nonempty support, a common positive weight scale cancels between numerator and denominator, including the factor four in Eq. equation 5. Each field remains a convex combination of its appearances. Their sum produces $\widehat{Y}_{\text{raw}} = Y_c + R_m + R_f$, as shown in Figure 2(a), allowing signed corrections outside the coarse field's colour convex hull. Rendering uses no depth ordering or transmittance product. Parameter mappings and the general accumulation form are given in Appendix A.1. Normalisation and CGW therefore play distinct roles: the former cancels common weight scaling, while the latter changes relative contributions. Signed appearance values provide the direction of the resulting correction.

3.6 Training and Full-Image Inference

The three fields are learned jointly from the composite reconstruction, without per-field targets. Training supervises the raw sum with pixelwise L1 and Fourier-domain L1 losses, corresponding to $\mathcal{L}_{\text{loss}}$ in Figure 2(a):

$$\mathcal{L}_{\text{loss}} = \mathcal{L}_1(\widehat{Y}_{\text{raw}}, Y) + 0.2\mathcal{L}_{\text{fft}}(\widehat{Y}_{\text{raw}}, Y). \tag{7}$$

The Fourier term compares real and imaginary transform components; both terms use mean reduction. Evaluation clips the sum to $[0, 1]$. At native image size, global features are cached and overlapping

crop predictions are combined by cosine blending. This bounds the active local field size, while total work grows with the number of crops. Loss definitions and the complete inference procedure appear in Appendix A.1 and Algorithm 1.

4 EXPERIMENTS

Datasets and evaluation. Task3 evaluates dehazing on SOTS from RESIDE (Li et al., 2019), deraining on Rain100L (Yang et al., 2017), and denoising on CBSD68 (Martin et al., 2001; Zhang et al., 2018) at $\sigma = 15, 25, 50$. Its average weights these five settings equally. Task5 uses denoising at $\sigma = 25$ and adds GoPro deblurring (Nah et al., 2017) and LOLv1 low-light enhancement (Wei et al., 2018), averaging five tasks equally. We report PSNR and SSIM (Wang et al., 2004). Downstream evaluation uses CamVid segmentation (Brostow et al., 2009), Penn–Fudan pedestrian detection (Wang et al., 2007), and NYUv2 depth and surface normals (Silberman et al., 2012). Dataset splits and task-specific metrics are given in Appendix D.1.

Implementation details. We train on 160×160 crops with AdamW (Loshchilov & Hutter, 2019) and the loss in Eq. equation 7. Full optimisation settings are in Appendix A.2. Published pixel-output results follow Tian et al. (2025); † marks the adaptations' original venues. Evaluation protocols are distinguished in Appendix B.2.

4.1 TASK3 AND TASK5 RECONSTRUCTION

GSAioIR achieves 32.1757 dB / 0.9143 SSIM on Task3 and 30.0232 dB / 0.9113 SSIM on Task5 (Tables 1 and 2). It exceeds both GaussianSR-R and the 9M GSASR-R restoration adaptations in mean PSNR and SSIM. The PSNR gains are 2.55 and 2.77 dB on Task3, and 4.22 and 4.83 dB on Task5, respectively. Higher PSNR across all displayed restoration settings supports the effectiveness of the shared Gaussian output interface across noise, haze, rain, blur, and low illumination.

The *dense-output* references predict pixel RGB without explicit Gaussian output fields. GSAioIR's Task3 average exceeds AirNet, Restormer, and PromptIR, but trails InstructIR and DFPIR. On Task5, it exceeds InstructIR's average and attains the highest listed GoPro PSNR, while DFPIR leads overall. Gaussian restoration is thus competitive with these pixel-output models, particularly in deblurring.

Deblurring may benefit from constrained Gaussian aggregation. Positive weighted mixing can smooth colour transitions, whereas signed residuals allow corrections beyond the coarse RGB average. CGW modulates overlap by content; together with sharp-target pixel and Fourier supervision, it may favour edge recovery. This interpretation remains a mechanism hypothesis rather than an isolated causal finding.

4.2 QUALITATIVE COMPARISON

Table 3: **Field removal on 668 crops.** Δ is removal minus full PSNR (dB); Better counts improvements after removing M. CBSD68 uses $\sigma = 25$.

Dataset	Mean Δ_M	Med. Δ_M	Better	Mean Δ_F
SOTS	-0.26	$+0.22$	277/500	-17.26
Rain100L	-2.33	-1.79	2/100	-23.02
CBSD68	-0.53	-0.37	0/68	-17.93

Figure 3 shows favourable restoration examples across five tasks. Distinct CBSD68 images illustrate three noise levels. Wall boundaries and thin branches expose differences in deraining and deblurring. Low-light predictions improve visibility but retain colour deviations. Methods share each row's observation and centre region, with full-image PSNR/SSIM. Gaussian models and DFPIR use five tasks, InstructIR seven; GSASR-R is the 9M adaptation. Protocols and fixed-index examples appear in Appendix B.2.

4.3 What the Gaussian fields represent

We examine one native 160×160 centre crop from every common-set image, with the correct crop box and context from the same full observation. A single forward stores the coarse RGB field C and residual fields M, F. Removing a field then tests the learned allocation without retraining. All 668 complete sums reproduce their stored predictions exactly. Table 3 summarises the changes after removal; full-crop PSNR and bootstrap intervals appear in Appendix C.1.

Table 3 and Figure 4 summarise these fixed-weight interventions.

Removing M lowers mean crop PSNR by 0.26 dB on SOTS, 2.33 dB on Rain100L, and 0.53 dB on CBSD68. Yet 277 of 500 SOTS crops improve, with a median change of $+0.22$ dB. The mean therefore hides a substantial image-dependent effect.

The fine field carries more spatial variation in these crops, but even removing F improves 15 SOTS examples. Figure 4 shows these distributions. These interventions make the learned allocation of corrections observable; the scales need not form fixed frequency bands.

The output interface also has a practical cost. Controlled profiling at 720×1280 records 266.7 MiB peak allocated memory and 1,245.9 ms for the tiled GSAioIR implementation, versus 8,386.3 MiB and 876.7 ms for full-image DFPIR. This is a trade-off between active memory and latency, not a representation-level speed advantage. Full profiling results are in Appendix C.2.

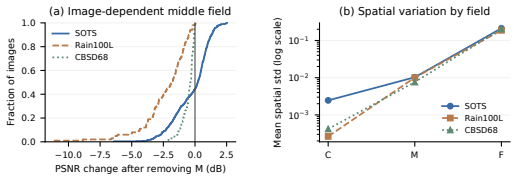

Figure 4: **Field behaviour.** Fixed-model analysis across the common cohort.

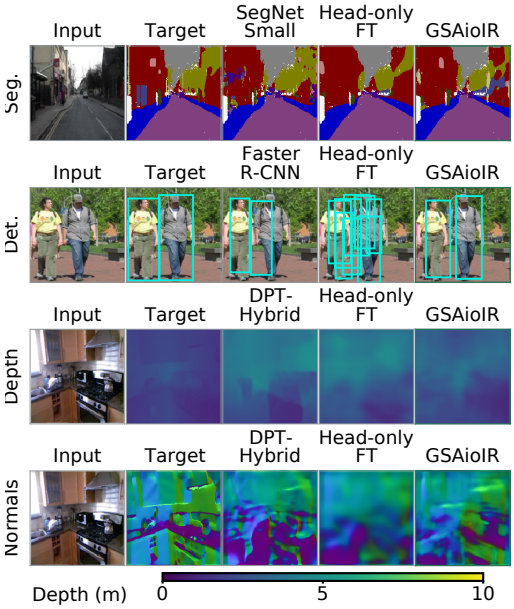

Figure 5: **Downstream predictions.** Shared display scales; boxes with scores ≥ 0.5.

4.4 Downstream transfer through full-model fine-tuning

We jointly fine-tune one backbone and task-specific Gaussian fields on mixed datasets: CamVid segmentation, Penn–Fudan detection, and NYUv2 depth and normals. Interleaved task-labelled microbatches require the shared backbone to accommodate dataset shifts and heterogeneous objectives. Segmentation and detection use separate fields; depth and normals share a field with independent readouts. All models receive clean 160×160 RGB inputs.

Table 4: **Downstream metrics.** Task-specific models: SegNetSmall (mIoU), Faster R-CNN MobileNetV3 (AP50), and separate DPT-Hybrid models (AbsRel/Angle).

Model	mIoU ↑	AP50 ↑	AbsRel ↓	Angle ↓
Task-specific	0.4034	0.3262	0.2959	35.94°
GSAioIR	**0.4658**	**0.8802**	**0.2702**	**31.23°**

Task-specific comparisons use SegNetSmall, a compact SegNet-style model (Badrinarayanan et al., 2017); Faster R-CNN (Ren et al., 2015) with MobileNetV3 (Howard et al., 2019); and separate DPT-Hybrid networks (Ranftl et al., 2021) for depth and normals, adapted from Omnidata (Eftekhar et al., 2021). Each is trained independently from scratch on its designated task and dataset. Splits, budgets and full metrics are in Appendix D.1.

GSAioIR performs better on all four metrics in Table 4. Compute, pretraining and supervision differ, so these gaps cannot be attributed solely to Gaussian outputs. Figure 5 also includes Head-only FT,

which trains only added fields and readouts with the backbone frozen. Its readouts and budget differ from full-model fine-tuning. The joint results support extending the Gaussian output interface to semantic and geometric prediction.

4.5 Component ablations and visual diagnostics

Table 5 compares Base model components on Task3. Training details are in Appendix C.1.

Table 5: **Component ablations.** PSNR/SSIM at the end of training under a shared protocol. CBSD68 averages $\sigma = 15, 25, 50$; Avg. retains equal weighting over the five original settings. C/M denotes coarse/middle. Bold marks the highest displayed PSNR.

Setting	Base model	No global context	No CGW	No fine skip	Single RGB field	Progressive fields	Head width 128	One C/M slot
SOTS	29.43/.957	28.59/.955	29.44/.958	29.57/.957	29.49/.957	29.44/.957	**29.60**/.958	29.34/.957
Rain100L	**36.83**/.976	36.61/.975	36.72/.976	36.66/.975	36.56/.975	36.75/.976	36.77/.976	36.72/.976
CBSD68	**31.07**/.874	31.00/.873	31.05/.874	31.04/.874	30.99/.873	31.05/.874	31.05/.873	31.03/.873
Avg.	31.89/.911	31.64/.910	31.86/.911	31.87/.911	31.80/.910	31.87/.911	**31.90**/.911	31.83/.911

Removing global conditioning decreases mean PSNR by 0.2498 dB, including 0.8432 dB on SOTS. Historical denoising validation also removes access to an independently corrupted global view, so this is not a pure architectural control under a single observation. Removing CGW and the fine skip changes the mean by -0.0330 and -0.0239 dB. A single RGB Gaussian field loses 0.0913 dB; progressive field conditioning loses 0.0253 dB. Reducing coarse/middle slots to one loses 0.0611 dB. Halving their width to 128 instead slightly improves PSNR but lowers SSIM. These observations do not support a general benefit from wider heads or every added component.

Figure 6 shows image-dependent effects. Removing global context or using a single RGB field improves the SOTS crop, whereas the Base model performs better on the rain crop. Full comparisons appear in Appendix C.1. These cases complement the validation means. Broader attribution to the Gaussian representation requires matched output-head and crop-geometry controls (Appendix C.1).

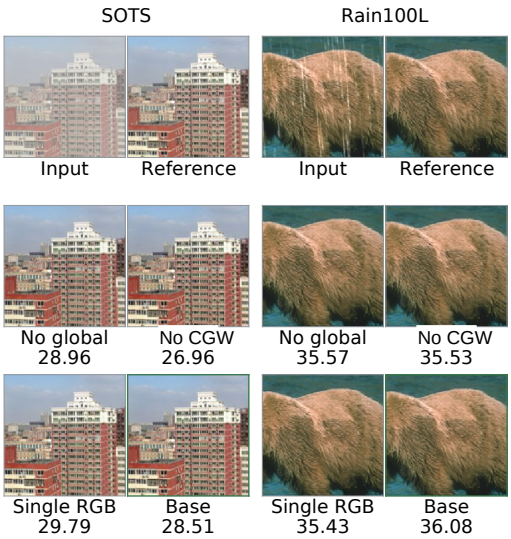

Figure 6: **Two ablation cases.** Crop PSNR (dB) at the end of training. Full grid: Fig. 17.

The retrained and fixed-weight analyses answer different questions. A retrained variant can redistribute information across its remaining fields, whereas removing a field from a trained model tests its existing allocation. Large removal losses therefore need not imply equally large retraining penalties. Together, the two analyses indicate that the usefulness of each field depends on both the learned decomposition and the restoration setting, rather than a fixed assignment of frequencies to scales. The small mean change without CGW is consistent with a refinement role within the full renderer.

5 Conclusion

GSAioIR uses independently normalised 2D Gaussian RGB and residual fields for five restoration tasks. It improves over the tested Gaussian adaptations and is competitive with pixel-output references. Mixed-task fine-tuning extends the backbone to segmentation, detection, depth, and normals. These results support Gaussian outputs across restoration and downstream vision, with a trade-off between active memory and latency.

AI USE STATEMENT

Generative AI tools assisted with drafting, literature and code checks, experiment scripting, execution, and analysis of recorded results. Numerical results were computed from model outputs or traceable experiment records. The authors are responsible for validating the manuscript and experiments.

REFERENCES

Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017. doi: 10.1109/tpami.2016.2644615.

Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, 2009. doi: 10.1016/j.patrec.2008.04.005.

Du Chen, Liyi Chen, Zhengqiang Zhang, and Lei Zhang. Generalized and efficient 2d gaussian splatting for arbitrary-scale super-resolution. In *IEEE/CVF International Conference on Computer Vision*, 2025. doi: 10.1109/iccv51701.2025.02453.

Yuhan Chen, Ying Fang, Guofa Li, Wenxuan Yu, Yicui Shi, Jingrui Zhang, Kefei Qian, Wenbo Chu, and Keqiang Li. LL-GaussianMap: Zero-shot low-light image enhancement via 2d gaussian splatting guided gain maps. *arXiv preprint arXiv:2601.15766*, 2026a. URL https://arxiv.org/abs/2601.15766.

Yuhan Chen, Wenxuan Yu, Guofa Li, Kunyang Huang, Ying Fang, Yicui Shi, Wenbo Chu, and Keqiang Li. Dehaze-GaussianImage: Zero-shot dehazing via efficient 2d gaussian splatting representation. *arXiv preprint arXiv:2606.16163*, 2026b. URL https://arxiv.org/abs/2606.16163.

Marcos V. Conde, Gregor Geigle, and Radu Timofte. Instructir: High-quality image restoration following human instructions. In *European Conference on Computer Vision*, 2024. doi: 10.1007/978-3-031-72764-1_1. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/html/5238_ECCV_2024_paper.php.

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. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/fb44a668c2d4bc984e9d6ca261262cbb-Abstract-Conference.html.

Ainaz Eftekhar, Alexander Sax, Jitendra Malik, and Amir Zamir. Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3D scans. In *IEEE/CVF International Conference on Computer Vision*, 2021. doi: 10.1109/iccv48922.2021.01061.

Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for MobileNetV3. In *IEEE/CVF International Conference on Computer Vision*, 2019. doi: 10.1109/iccv.2019.00140. URL https://openaccess.thecvf.com/content_ICCV_2019/html/Howard_Searching_for_MobileNetV3_ICCV_2019_paper.html.

Jintong Hu, Bin Xia, Bin Chen, Wenming Yang, and Lei Zhang. Gaussiansr: High fidelity 2d gaussian splatting for arbitrary-scale image super-resolution. In *AAAI Conference on Artificial Intelligence*, 2025. doi: 10.1609/aaai.v39i4.32369.

Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. doi: 10.1109/cvpr.2015.7299156.

Jung In Jang and Kyong Hwan Jin. GRAPE (gaussian rendering for accelerated pixel enhancement) brings fast and lightweight arbitrary super-resolution. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2026. doi: 10.1109/wacv61042.2026.00748.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. doi: 10.1145/3592433. URL https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/.

Boyi Li, Wenqi Ren, Dengpan Fu, Dacheng Tao, Dan Feng, Wenjun Zeng, and Zhangyang Wang. Benchmarking single-image dehazing and beyond. *IEEE Transactions on Image Processing*, 28 (1):492–505, 2019. doi: 10.1109/tip.2018.2867951.

Boyun Li, Xiao Liu, Peng Hu, Zhongqin Wu, Jiancheng Lv, and Xi Peng. All-in-one image restoration for unknown corruption. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. doi: 10.1109/cvpr52688.2022.01693.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, 2014. doi: 10.1007/978-3-319-10602-1_48.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017a. doi: 10.1109/cvpr.2017.106.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision*, 2017b. doi: 10.1109/iccv.2017.324.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Kede Ma, Zhengfang Duanmu, Qingbo Wu, Zhou Wang, Hongwei Yong, Hongliang Li, and Lei Zhang. Waterloo Exploration Database: New challenges for image quality assessment models. *IEEE Transactions on Image Processing*, 26(2):1004–1016, 2017. doi: 10.1109/tip.2016.2631888.

David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *IEEE International Conference on Computer Vision*, volume 2, pp. 416–423, 2001. doi: 10.1109/iccv.2001.937655. URL https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/papers/mftm-iccv01.pdf.

Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In *International Conference on 3D Vision*, 2016. doi: 10.1109/3dv.2016.79.

Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. doi: 10.1109/cvpr.2017.35.

Long Peng, Anran Wu, Wenbo Li, Peizhe Xia, Xinjie Zhang, Xueyuan Dai, Xin Di, Haoze Sun, Renjing Pei, Yang Wang, Yang Cao, and Zheng-Jun Zha. Pixel to gaussian: Ultra-fast continuous super-resolution with 2d gaussian modeling. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/94ccfdb2ca14f33a86a0b9b7d0c1bfb1-Abstract-Conference.html.

Vaishnav Potlapalli, Syed Waqas Zamir, Salman Khan, and Fahad Shahbaz Khan. Promptir: Prompting for all-in-one image restoration. In *Advances in Neural Information Processing Systems*, 2023. doi: 10.52202/075280-3121.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *IEEE/CVF International Conference on Computer Vision*, 2021. doi: 10.1109/iccv48922.2021.01196.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015. URL https://papers.nips.cc/paper/2015/hash/14bfa6bb14875e45bba028a21ed38046-Abstract.html.

Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. doi: 10.1109/cvpr.2019.00075.

David Serrano-Lozano, Luis Herranz, Shaolin Su, and Javier Vazquez-Corral. Adaptive blind all-in-one image restoration. *Computer Vision and Image Understanding*, 269:104795, 2026. doi: 10.1016/j.cviu.2026.104795. Comparison values taken from arXiv:2411.18412v2, Table 1.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *European Conference on Computer Vision*, 2012. doi: 10.1007/978-3-642-33715-4_54.

Xiangpeng Tian, Xiangyu Liao, Xiao Liu, Meng Li, and Chao Ren. Degradation-aware feature perturbation for all-in-one image restoration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. doi: 10.1109/cvpr52734.2025.02623.

Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully convolutional one-stage object detection. In *IEEE/CVF International Conference on Computer Vision*, 2019. doi: 10.1109/iccv.2019.00972.

Liming Wang, Jianbo Shi, Gang Song, and I-fan Shen. Object detection combining recognition and segmentation. In *Asian Conference on Computer Vision*, 2007. doi: 10.1007/978-3-540-76386-4_17.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13 (4):600–612, 2004. doi: 10.1109/tip.2003.819861. URL https://ece.uwaterloo.ca/~z70wang/publications/ssim.html.

Chen Wei, Wenjing Wang, Wenhan Yang, and Jiaying Liu. Deep retinex decomposition for low-light enhancement. In *British Machine Vision Conference*, 2018. URL https://bmva-archive.org.uk/bmvc/2018/contents/papers/0451.pdf.

Wenhan Yang, Robby T. Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan. Deep joint rain detection and removal from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. doi: 10.1109/cvpr.2017.183.

Eduard Zamfir, Zongwei Wu, Nancy Mehta, Yuedong Tan, Danda Pani Paudel, Yulun Zhang, and Radu Timofte. Complexity experts are task-discriminative learners for any image restoration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. doi: 10.1109/cvpr52734.2025.01190.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. doi: 10.1109/cvpr52688.2022.00564.

Kai Zhang, Wangmeng Zuo, and Lei Zhang. FFDNet: Toward a fast and flexible solution for CNN-based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, 2018. doi: 10.1109/tip.2018.2839891.

Xinjie Zhang, Xingtong Ge, Tongda Xu, Dailan He, Yan Wang, Hongwei Qin, Guo Lu, Jing Geng, and Jun Zhang. Gaussianimage: 1000 fps image representation and compression by 2d gaussian splatting. In *European Conference on Computer Vision*, 2024. doi: 10.1007/978-3-031-72673-6_18. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/html/1421_ECCV_2024_paper.php.

SUPPLEMENTARY MATERIAL

A IMPLEMENTATION AND TRAINING

A.1 ARCHITECTURE AND GAUSSIAN SYNTHESIS

Table 6: GSAioIR configurations for 160×160 crops. Head widths are channel counts.

Setting	Base	Expanded
Encoder depths	$(2, 3, 3)$	$(2, 3, 4)$
HGPD blocks	4	6
Coarse head width	256	256
Middle head width	256	320
Middle refinement blocks	0	2
Fine slots per cell	8	12
Slot dimension	80	80
Slot interaction blocks	3	3
Coarse primitives	1,600	1,600
Middle primitives	6,400	6,400
Fine primitives	12,800	19,200
Total primitives	20,800	27,200

The Base model and expanded configurations share the same global-to-local prediction and three-field rendering formulation. They differ in capacity, including the number of fine slots, decoder depth, and middle-head processing. The expanded configuration is used for the restoration comparisons, while the Base model is used for component ablations. Table 6 summarises both configurations. The expanded model contains 27,847,797 parameters.

Both configurations use an 832-pixel long-size budget for the bounded global view, a 16-pixel global patch size, at most 512 global tokens, and masked padding. Four global self-attention blocks encode the view. These settings bound context processing; canvas constraints can affect unusual aspect ratios. There is no separately rendered global Gaussian bank.

Figure 2 uses feature names based on spatial scale. For a 160×160 crop, F_{high}, F_{middle}, and F_{low} have spatial sizes 80×80, 40×40, and 20×20, with 72, 144, and 192 channels, respectively. HGPD denotes the Hierarchical Gaussian Physical Decoder. Its output is the latent feature F_{fused}, with 320 channels at 40×40 resolution.

HGPD first applies a separate 1×1 projection to each local feature. Area resizing reduces the high-resolution source to the middle grid; bilinear interpolation enlarges the low-resolution source. Their mean initialises the decoder state. Each decoder block transforms the three sources with separate pointwise-convolution/GELU adapters and mixes them with learned softmax weights. The spatial condition passes through a pointwise-convolution/GELU/pointwise-convolution adapter. The block then applies affine modulation from the concatenated task modulation and learned degradation-prior vectors.

For global summary attention, the local decoder state supplies queries, while normalised and projected summary tokens supply keys and values. Eight-head cross-attention updates the local state. The convolutional feed-forward block uses group normalisation, a pointwise expansion, a 3×3 depthwise convolution, a SiLU gate, and a pointwise output projection. The source, condition, attention and local updates are residual and use learned scale parameters initialised to 10^{-2}. A final group normalisation produces F_{fused}. The illustrated configuration has four decoder blocks and the expanded configuration six. This is a feature-fusion decoder; the subsequent heads perform Gaussian parameter readout.

Figure 7 illustrates why normalisation and signed residual composition address different aspects of Gaussian aggregation: the former removes common weight-scale dependence, while the latter permits corrections beyond the coarse field's colour hull.

Expanded query/key construction. The pixel queries and primitive keys used in Eq. (5) are formed from projection logits with a positive-simplex mapping. For the slot tensor S, the full operations are

$$\begin{aligned} \overline{S}_{uv} &= \frac{1}{K_f} \sum_{j=1}^{K_f} S_{uvj}, \\ \ell_q(p) &= \mathrm{up}[W_q \overline{S}](p), \qquad \ell_{k,i} = W_k S_i, \\ \pi_\varepsilon(z) &= (1-\varepsilon)\,\mathrm{softmax}(z) + \frac{\varepsilon}{4}\mathbf{1}, \qquad \varepsilon = 0.01. \end{aligned} \tag{8}$$

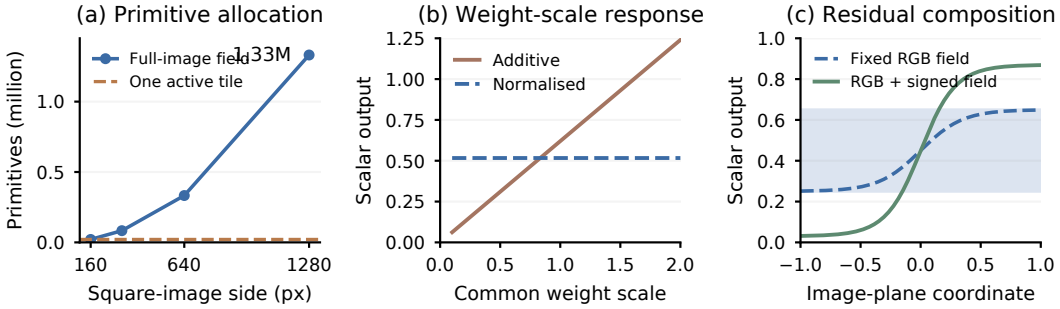

Figure 7: **Design issues in Gaussian restoration.** (a) Primitive count at fixed Base model density grows with image area; a 160-pixel crop bounds only the active local allocation. (b) Scaling fixed positive weights changes additive output but not normalised output. (c) A signed correction can leave a fixed coarse field's colour hull (shaded). Panels (b,c) are analytical scalar examples, not model measurements.

W_q and W_k denote learned affine projections. Query logits are bilinearly upsampled before softmax. Guidance uses shared slot features before the separate geometry and appearance task modulation. The four components determine the factor four in Eq. (5); CGW denotes content-guided weighting. When the query is uniform, that factor times the query/key inner product equals one, recovering ordinary Gaussian weighting.

Prediction-head implementation. The coarse and middle heads in Eq. (3) apply a pointwise convolution, GELU, a 3×3 depthwise convolution, GELU, and another pointwise convolution and GELU before parameter readout. The expanded middle head adds two residual refinement blocks. The fine head uses three slot-interaction blocks, followed by feature modulation and separate task-modulated geometry and appearance translators. The remaining implementation details specify primitive mappings, finite rendering support and crop synthesis.

Gaussian parameters and appearance mappings. The primitive parameters and covariance matrix are

$$\gamma_i = (\mu_i, \sigma_{x,i}, \sigma_{y,i}, \rho_i, \alpha_i, c_i), \qquad \Sigma_i = \begin{pmatrix} \sigma_{x,i}^2 & \rho_i \sigma_{x,i}\sigma_{y,i} \\ \rho_i \sigma_{x,i}\sigma_{y,i} & \sigma_{y,i}^2 \end{pmatrix}. \tag{9}$$

Centres and RGB appearances contain two and three scalars, respectively. Positive scales and correlation magnitude below one make the covariance positive definite. Appearance logits are mapped as

$$c_i^c = \mathrm{sigmoid}(z_i^c), \qquad c_i^m = \tanh(z_i^m), \qquad c_i^f = 2\,\mathrm{sigmoid}(z_i^f) - 1. \tag{10}$$

These mappings give coarse RGB and signed middle/fine corrections without negative Gaussian amplitudes. The spatial kernel is defined by Eq. (A) in Figure 2(d); coarse and middle weights are $w_i^s(p) = \alpha_i^s G_i^s(p)$, with fine guidance specified in Eq. (5).

Kernel values at most 10^{-6} are discarded before amplitude weighting. The cutoff does not depend on a common amplitude scale, so uniform weight-rescaling invariance holds on fixed nonempty support.

General form of the figure's rendering equations. Eq. (B) in Figure 2(d) illustrates three overlapping primitives. For an arbitrary field $s \in \{c, m, f\}$ with N_s primitives, the appearance numerator and weight density are

$$A_s(p) = \sum_{i=1}^{N_s} w_i^s(p) c_i^s, \qquad D_s(p) = \sum_{i=1}^{N_s} w_i^s(p). \tag{11}$$

Weights outside retained support are zero. Independent normalisation and composition give

$$\mathcal{R}_s(p) = \begin{cases} A_s(p)/D_s(p), & D_s(p) > 0, \\ 0, & D_s(p) = 0, \end{cases} \qquad \widehat{Y}_{\mathrm{raw}}(p) = Y_c(p) + R_m(p) + R_f(p). \tag{12}$$

Here $Y_c = \mathcal{R}_c$, $R_m = \mathcal{R}_m$, and $R_f = \mathcal{R}_f$; the figure's $I(p)$ denotes the corresponding single-field output. Figure 2(a) shows the raw composite prediction $\widehat{Y}_{\text{raw}}$ and its clean target Y. The displayed feature grids and slot counts describe the 160×160 Base model crop configuration. Native full-image inference uses the crop synthesis and blending procedure below.

Positive query/key mapping. Applying the floor mapping to the logits in Eq. (8) gives

$$q(p) = \pi_\varepsilon(\ell_q(p)), \qquad k_i = \pi_\varepsilon(\ell_{k,i}). \tag{13}$$

The floor weight is $\varepsilon = 0.01$. The resulting four-component vectors are positive and sum to one.

Loss and synthesis details. Writing $\mathcal{F}$ for the orthonormal two-dimensional Fourier transform, the complete loss in Eq. (7) is

$$\mathcal{L}_{\text{loss}} = \|\widehat{Y}_{\text{raw}} - Y\|_1 + 0.2\| \operatorname{ReIm} \mathcal{F}(\widehat{Y}_{\text{raw}}) - \operatorname{ReIm} \mathcal{F}(Y)\|_1. \tag{14}$$

ReIm stacks real and imaginary components. Both losses use mean reduction and supervise only the unclipped composite prediction. No per-field target or fixed frequency partition is imposed. The global features are computed once per full image. Each local crop uses its normalised full-image box, and native-resolution predictions are combined with cosine blending weights. Active local field size is bounded, while total inference work grows with the number of crops.

Coarse and middle heads initialise four separated offsets within each reference cell. For grid dimensions (H_s, W_s), the cell anchor is $a_{uv} = ((v + 1/2)/W_s, (u + 1/2)/H_s)$. If o_j is a fixed within-cell offset, the predicted centre in edge-normalised coordinates is

$$e_{uvj} = \operatorname{clip}_{[0,1]} \left[a_{uv} + \left(o_j + 0.2 \tanh z^{\mu}_{uvj}\right) \odot (W_s^{-1}, H_s^{-1}) \right]. \tag{15}$$

The conversion to the rasteriser's align-corners coordinates is

$$\phi_{H,W}(e) = \left((2e_x - 1)\frac{W}{W-1}, (2e_y - 1)\frac{H}{H-1} \right). \tag{16}$$

At the 160-pixel reference size, the coarse and middle scale ranges are $(0.035, 0.10)$ and $(0.0125, 0.04)$, respectively. For a dimension $D \in \{H, W\}$, their scale mapping is $(\sigma_{\min} + (\sigma_{\max} - \sigma_{\min}) \operatorname{sigmoid} z^{\sigma}) 159/(D-1)$. Correlation is $0.8 \tanh z^{\rho}$ and weight amplitude is $0.1 + 0.9 \operatorname{sigmoid} z^{\alpha}$.

The fine head uses bounded, task-modulated geometry and appearance. On a grid of size (H_s, W_s), let a_{uv} be the cell anchor and z^{μ}_{uvj} the learned shift logits. A fixed slot initialisation z^{init}_j and temperature three give

$$e^{f}_{uvj} = \operatorname{clip}_{[0,1]} \left[a_{uv} + 3 \tanh\left(z^{\text{init}}_j + z^{\mu}_{uvj}/3\right) \odot (W_s^{-1}, H_s^{-1}) \right]. \tag{17}$$

Centres are converted by Eq. equation 16. For each axis, define $r = \operatorname{sigmoid}(z^{\sigma})$ and $u = \operatorname{clip}_{[0,1]}((r - 0.75)/0.25)$. The fine scale is

$$\sigma_f = \left[0.005 \exp(r \log 20) + 0.1u^2(3 - 2u)\right] \frac{159}{D-1}, \qquad D \in \{H, W\}. \tag{18}$$

At the reference size, the range is $(0.005, 0.2)$, so a fine primitive is not required to have narrower support than every coarse primitive. Fine correlation is $0.999 \tanh z^{\rho}$; its weight amplitude uses sigmoid with numerical clipping before the fixed slot-role transform. The released configuration sets the role-logit span to zero. Fine appearance is twice sigmoid minus one. Guidance is computed from shared slot features before separate geometry and appearance task modulation.

All sums use finite support and discard kernel values $G_i(p) \le 10^{-6}$ before multiplying by the weight amplitude. This threshold is independent of a common amplitude scale. Coarse appearance uses sigmoid and middle appearance uses tanh; all fields retain positive aggregation weights.

Algorithm 1 gives the formal parallel synthesis path. No rendered field is used to condition another field's prediction. The primitive sets are nevertheless jointly trained through shared features and a common composite-image loss.

N/D divides exactly where accumulated density is positive and returns zero otherwise. The common positive weight-rescaling invariance applies to the retained supports. It does not imply invariance to moving, adding, or removing individual primitives. The renderer uses no depth coordinate, visibility ordering, or transmittance product.

Algorithm 1 Task-conditioned multiscale Gaussian crop synthesis

Require: Degraded crop X^k, cached global features Z^g, crop box b^k, task t, fine slot count K_f
Ensure: Restored crop $\widehat{Y}^k$

Encode task condition and local features $F_{\text{high}}, F_{\text{middle}}, F_{\text{low}}$
Retrieve crop-aligned global context using Z^g and b^k
Fuse local and global features into F_{fused}
$\{\gamma_i^c\}_{i=1}^{N_c} \leftarrow \textsc{CoarseHead}([F_{\text{low}}, \text{pool}(F_{\text{fused}})])$
$\{\gamma_i^m\}_{i=1}^{N_m} \leftarrow \textsc{MiddleHead}([F_{\text{middle}}, F_{\text{fused}}])$
$H_f \leftarrow F_{\text{fused}} + P_{\downarrow}(F_{\text{high}})$
$(\{\gamma_i^f\}_{i=1}^{N_f}, S) \leftarrow \textsc{FineSlotHead}(H_f, t, K_f)$
Form four-component pixel queries q and primitive keys k from S
$Y_c \leftarrow \text{NDRender}(\{\gamma_i^c\}_{i=1}^{N_c})$; $R_m \leftarrow \text{NDRender}(\{\gamma_i^m\}_{i=1}^{N_m})$
$R_f \leftarrow \textsc{GuidedNDRender}(\{\gamma_i^f\}_{i=1}^{N_f}, q, k)$
$\widehat{Y}^k_{\text{raw}} \leftarrow Y_c + R_m + R_f$
return raw sum during training; $\text{clip}_{[0,1]}(\widehat{Y}^k_{\text{raw}})$ at evaluation

A.2 Training data and optimisation

Training uses OTS from RESIDE (Li et al., 2019), RainTrainL (Yang et al., 2017), BSD400 (Martin et al., 2001), Waterloo Exploration (Ma et al., 2017), GoPro (Nah et al., 2017), and LOLv1 (Wei et al., 2018). Table 7 summarises the training data and sampling configuration. Counts refer to unique input/target pairs or clean images before repetition. Denoising reuses the same 5,144 clean images at each noise level. Repetition balances the task mixture; it does not create additional independent images. The resulting configured epoch contains 142,431 entries for Task3 and 200,973 for Task5.

Table 7: Training file-list inventory and configured repetition factors. Deblurring and low-light data are included only in Task5.

Task	Source	Unique entries	Repetitions	Epoch entries
Dehazing	OTS	72,135	1	72,135
Denoising, each σ	BSD400 + WaterlooED	5,144	3	15,432
Deraining	RainTrainL	200	120	24,000
Deblurring	GoPro	2,103	14	29,442
Low-light	LOLv1	485	60	29,100

The released Task3 and Task5 recipes use 80 epochs, batch size 32 across two GPUs, and AdamW with $\beta = (0.9, 0.999)$ and weight decay 10^{-4}. The learning rate warms from 10^{-4} to 3×10^{-4} over 20 epochs and follows cosine decay to 10^{-5}. Gradient norms are clipped at three. This describes the configured schedule; the selected model updates are reported in Table 10. The component study uses a shorter schedule, specified in Appendix C.1.

Paired images undergo the same horizontal and vertical flips, followed by a common random 160×160 crop. The global view is built from the corresponding full degraded image, and the crop box records its normalised location. For denoising training, Gaussian noise is added to the full clean image at $\sigma \in \{15, 25, 50\}$, clipped and rounded to uint8, and then used for both the local crop and global view. This training construction uses one noisy observation; the historical evaluation difference is analysed separately in Appendix C.2.

Restoration adaptations. GaussianSR-R and GSASR-R train separate Task3/Task5 models from scratch for 40,000 updates, with seed 3402, 160×160 crops, batch size 32 on two RTX 4090 GPUs, AdamW, and the loss in Eq. equation 7. A 1,000-step warmup precedes the 10^{-4} peak learning rate and cosine decay to 10^{-6}. Final-update evaluation uses RGB uint8 metrics, zero border crop, 160-pixel tiles, 32-pixel overlap, and fixed noise. Neither adaptation receives task labels or global context. Their smaller capacities and shorter budgets distinguish these experiments from the released GSAioIR training recipe.

Table 8: Additional published Task3 references not listed in Table 1. PSNR/SSIM and source-reported five-setting means follow Tian et al. (2025). †The inconsistent DL SSIM values are retained as reported.

Method	Venue	Dehazing SOTS	Deraining Rain100L	Denoising: CBSD68 $\sigma = 15$	Denoising: CBSD68 $\sigma = 25$	Denoising: CBSD68 $\sigma = 50$	Avg-5
DL†	TPAMI'21	26.92/.391	32.62/.931	33.05/.914	30.41/.861	26.90/.740	29.98/.875
MPRNet	CVPR'21	25.28/.954	33.57/.954	33.54/.927	30.89/.880	27.56/.779	30.17/.899

Table 9: Additional published Task5 references not listed in Table 2. PSNR/SSIM follow the original protocols; denoising uses $\sigma = 25$ and Avg-5 weights the five tasks equally. Sources are Tian et al. (2025) and, for $*$ rows, Serrano-Lozano et al. (2026).

Method	Venue	Dehazing SOTS	Deraining Rain100L	Denoising CBSD68	Deblurring GoPro	Low-light LOLv1	Avg-5
DGUNet	CVPR'22	24.78/.940	36.62/.971	31.10/.883	27.25/.837	21.87/.823	28.32/.891
SwinIR	ICCVW'21	21.50/.891	30.78/.923	30.59/.868	24.52/.773	17.81/.723	25.04/.835
NAFNet	ECCV'22	25.23/.939	35.56/.967	31.02/.883	26.53/.808	20.49/.809	27.76/.881
DL	TPAMI'21	20.54/.826	21.96/.762	23.09/.745	19.86/.672	19.83/.712	21.05/.743
Transweather	CVPR'22	21.32/.885	29.43/.905	29.00/.841	25.12/.757	21.21/.792	25.22/.836
PromptIR*	NeurIPS'23	28.26/.965	35.40/.967	30.89/.872	26.55/.808	21.80/.815	28.58/.885
DiffUIR*	CVPR'24	28.17/.964	35.52/.969	30.92/.879	26.99/.821	20.92/.789	28.50/.880
X-Restormer*	ECCV'24	27.58/.959	35.42/.968	30.92/.880	27.54/.835	20.88/.817	28.47/.891
AdaIR*	ICLR'25	30.53/.978	38.02/.981	31.35/.889	28.12/.858	23.00/.845	30.20/.910
ABAIR-OH*	CVIU'26	33.46/.983	38.18/.983	31.38/.898	29.00/.878	24.20/.865	31.24/.921
ABAIR-SW*	CVIU'26	33.48/.984	38.22/.984	31.38/.898	29.00/.878	24.19/.865	31.25/.921

B Restoration evaluation

B.1 Model records and averaging conventions

The main restoration tables use separate Task3 and Task5 models and retain their original PSNR/SSIM summaries. The Task3 mean covers SOTS, Rain100L and three CBSD68 noise levels; the Task5 mean covers five tasks with denoising at $\sigma = 25$. Both average five settings equally.

The original denoising evaluation constructs the global noisy view separately from the local observation. We distinguish these results from the shared-observation benchmarks below, where every model receives the same degraded image and GSAioIR derives global context from that image. The full-resolution Rain100L field example in Appendix C.1 uses the Task3 model; the shared-observation comparisons and complete-cohort field analysis use the five-task model.

Historical reference values. These tables add published methods omitted from Tables 1 and 2, without repeating their comparison rows. They are distinct from the shared-observation evaluation in Appendix B.2. The Task3 Avg-5 includes three noise levels; it must not be compared directly with Macro-3. Values from different sources retain their own training and metric protocols.

Table 10: GSAioIR model selection. Entries are update/PSNR; last denotes the last recorded validation.

Task	Selected	Last
Task3	249256/32.1757	356080/32.1639
Task5	251200/30.0232	282600/30.0096

Table 11: Gaussian adaptations. Parameter counts and source-reported mean PSNR use the aggregation rules defined in the text.

Setting	GaussianSR-R	GSASR-R
Parameters (M)	1.838	9.264
Task3 Avg-3	29.1746	29.3518
Task3 Avg-5	29.6289	29.4099
Task5 Avg-5	25.8027	25.1957
Task5 Avg-7	27.0157	26.0326

Table 12: Additional Task5 denoising results and Avg-7 (PSNR/SSIM). The extra noise levels do not enter the Task5 Avg-5.

Setting	GaussianSR-R	GSASR-R
$\sigma = 15$	33.130/0.9200	29.762/0.8737
$\sigma = 50$	26.966/0.7465	26.487/0.7379
Avg-7	27.0157/0.8393	26.0326/0.8201

Result sources and aggregation. Table 10 records the selected Task3 and Task5 training updates. The expanded configuration contains 27.85M parameters; the Base configuration is reserved for the component study.

For Gaussian adaptations, Task3 Avg-5 uses SOTS, Rain100L, and three noise levels. The source's Avg-3 omits noise $\sigma = 15, 50$. Task5 Avg-5 adds GoPro and LOLv1 to Avg-3; Avg-7 also includes the other two noise levels.

The historical-reference tables retain source-reported means at their reported precision. The Gaussian adaptation means come from the original adaptation evaluation; GSAioIR values use the corresponding result summary. These records are kept separate from the shared-observation re-evaluation.

Published comparison rows are transcribed from Tables 1 and 2 of Tian et al. (2025), including DFPIR itself. Additional five-task results follow Table 1 of the second arXiv version of Serrano-Lozano et al. (2026). Reference methods are included for architectural coverage, with no upper score threshold. The Task3 DL row reports SOTS SSIM 0.391 and average SSIM 0.875, which are inconsistent with the arithmetic mean of its displayed values (0.7674). Both are retained with a dagger marker in Table 8. Other source means can differ from recomputation after rounding; these discrepancies are recorded in the numerical audit.

B.2 Shared-observation benchmarks

The common manifest contains all 500 SOTS pairs, 100 Rain100L pairs, and 68 CBSD68 observations at $\sigma = 25$. File names, dimensions, decoded input tensors, and target RGB arrays are checked for every entry. The full five-task manifest adds 1,111 GoPro pairs and 15 LOLv1 pairs. Urban100 (Huang et al., 2015) also contains 100 available clean test images, but is outside these two declared result tables. The local SOTS images are consistent with an outdoor split; an independent checksum of the original official archive has not been established. The exact evaluated file manifest is therefore retained as part of the experiment record.

Table 13: Scope of the original four-model contextual comparison. Counts refer to evaluated images, not the number of displayed examples.

Model	Common set	Five-task set	Supplied condition
InstructIR-7D	668	1,794	Fixed task instruction
PromptIR-3D	668	Not applicable	None
DFPIR-5D	668	1,794	Fixed degradation text
GSAioIR-5D	668	1,794	Task label and observed context

The public InstructIR release contains seven-task weights, not the five-task model underlying the historical reference row. The available PromptIR release is a three-task model; it is not the separate five-task reproduction labelled PromptIR* in the reference table. DFPIR uses the official five-task weight file. All active model parameters are loaded for evaluation. The model identities and loading checks are retained in the accompanying experiment records.

For InstructIR, instructions are fixed per task before scoring: "Remove the haze in the image", "Remove the rain in the image", "Remove the noise in the image", "Remove the blur in the image", and "fix the illumination". DFPIR uses the corresponding degradation strings from its official evaluation script. No instruction is selected from a per-image search. PromptIR receives no task label.

The common test uses one saved FP32 observation per image for every model. CBSD68 uses $X = \mathrm{clip}_{[0,1]}(Y + \epsilon)$ with independent Gaussian entries $\epsilon \sim \mathcal{N}(0, (25/255)^2)$ and seed 93404. No intermediate uint8 quantisation is applied to this saved observation. GSAioIR reconstructs its global view from that saved noisy observation, rather than independently corrupting a clean resized target. InstructIR uses its native padding; PromptIR and DFPIR reflect-pad to multiples of eight and trim the output. This differs from DFPIR's original centre-crop preprocessing, so the table is a common-input model comparison rather than a reproduction of published protocol-specific numbers. InstructIR, PromptIR, and DFPIR run full-image FP32 without resizing, ensembling, or OOM tiling fallback. GSAioIR uses 160-pixel tiles, overlap 32, and tile batch size one.

Dataset scores are arithmetic means of per-image RGB uint8 metrics. Macro-3 and Macro-5 then weight the designated datasets equally, not by image count. Paired confidence intervals condition on fixed models. The 500 SOTS filenames contain 492 distinct scene prefixes; resampling uses these groups and individual images for Rain100L and CBSD68. The comparison uses 5,000 bootstrap replicates with a fixed seed. GoPro frames are not treated as independent samples for significance claims. No interval here measures variation over training seeds.

Expanded noise-level evaluation. The main tables use separately trained Task3 and Task5 models, with selected updates recorded in Table 10. Task3 averages SOTS, Rain100L, and the three CBSD68 noise levels; Task5 averages SOTS, Rain100L, CBSD68 at $\sigma = 25$, GoPro, and LOLv1. Thus both means contain five terms but do not represent the same evaluation. The three-setting Gaussian adaptation summary instead uses three terms. We use its five-term full-validation averages in Table 1, not those incompatible headline averages.

Aggregate results retain their reported precision and are not recomputed from rounded per-setting values. Historical GSAioIR denoising used independently corrupted global context. Neither literature values nor these original records should be interpreted as results of the later shared-input protocol. Figure captions identify the models and observation protocols used for qualitative comparisons and fixed-weight interventions.

For a separate controlled-observation analysis, the complete common suite contains 500 SOTS, 100 Rain100L and 68 CBSD68 images. Evaluating CBSD68 at three noise levels gives 804 observations; the clean-image count remains 668. GSAioIR and the two Gaussian adaptation rows below retain their five-task models. AirNet and PromptIR retain three-task models, InstructIR seven-task, and DFPIR five-task. This is an evaluation scope, not a claim of identical training scopes.

The additional $\sigma = 15, 50$ evaluations cover all 68 CBSD68 images with the same seven models and saved FP32 observations. Noise seeds are 93403 and 93405, respectively. GSAioIR derives its global view from the same observation. Its five-task model obtains 31.89575 dB over the five settings, compared with the Task3-specific model's original 32.1757 dB. These values describe different model and observation protocols.

Published references were checked against Tables 1 and 2 of Tian et al. (2025). They remain separate from shared-observation results and retain the source averages. The inconsistent historical DL row is not included in the main tables. The compact ablation table averages three CBSD68 noise columns for display only; its original five-setting mean is unchanged. Table 17 supplies the individual noise-level values in Appendix C.1.

Table 14 reports the five-setting comparison and retains Avg-3 to make the aggregation difference explicit.

Archived predictions. The five-task comparison combines new predictions with existing GSAioIR GoPro and LOLv1 outputs from the same five-task model. All 1,126 existing outputs are re-scored against the current targets. Their recorded tiling configuration is 160-pixel crops with 32-pixel overlap.

Table 14: **Shared-observation evaluation across five settings.** PSNR/SSIM on 804 observations. Avg-5 weights all five settings equally; Avg-3 uses SOTS, Rain100L and CBSD68 at $\sigma = 25$. T denotes training scope, and † a five-task restoration adaptation. Bold marks column maxima within the measured comparison; training budgets differ.

Method	SOTS	Rain100L	CBSD68			Avg-5	Avg-3
			$\sigma = 15$	$\sigma = 25$	$\sigma = 50$		
Gaussian implementations: shared-observation evaluation							
GaussianSR-R†	24.08/0.892	30.47/0.935	33.13/0.920	30.40/0.864	26.97/0.747	29.01/0.871	28.32/.897
GSASR-R†	25.95/0.926	27.15/0.850	29.75/0.873	28.73/0.832	26.49/0.738	27.62/0.844	27.28/.869
GSAioIR (5T)	**29.73/0.958**	36.65/0.976	33.84/**0.933**	31.23/0.888	28.03/**0.798**	31.90/0.910	32.54/.9404
Dense-output models: shared-observation evaluation							
AirNet (3T)	27.21/0.943	34.89/0.967	33.87/0.932	31.24/0.888	28.00/0.797	31.04/0.905	31.11/.933
PromptIR (3T)	28.88/**0.958**	37.42/**0.978**	**33.91/0.933**	**31.28**/0.888	28.03/0.796	31.90/**0.911**	32.53/**.9413**
InstructIR (7T)	26.71/0.942	35.84/0.970	33.87/**0.933**	31.22/**0.889**	27.93/0.797	31.11/0.906	31.26/.9334
DFPIR (5T)	29.38/**0.958**	**37.62/0.978**	33.88/0.932	31.20/0.887	**28.04**/0.796	**32.02**/0.910	**32.74**/.9408

The original inference precision was not recorded. An FP32 check at the first, middle and last indices of each dataset differs by at most one uint8 level, 0.012758 dB PSNR and 0.000171 SSIM. This supports numerical agreement on those six cases; the full-set results use the re-scored outputs. Per-image metrics and model provenance are retained with the experiment records.

Table 15 extends the shared-observation comparison to blur and low light, retaining the archive checks described above.

Table 15: **Shared-observation five-task evaluation.** PSNR/SSIM on 1,794 inputs per model; Avg-5 weights tasks equally, with CBSD68 at $\sigma = 25$. T and † follow Table 14. GSAioIR GoPro/LOLv1 use archived predictions independently re-scored in full. Bold marks column maxima within the measured comparison.

Method	SOTS	Rain100L	CBSD68	GoPro	LOLv1	Avg-5
Gaussian implementations: shared-observation evaluation						
GaussianSR-R†	24.08 / 0.892	30.47 / 0.935	30.40 / 0.864	25.49 / 0.778	18.56 / 0.740	25.80 / 0.842
GSASR-R†	25.95 / 0.926	27.15 / 0.850	28.73 / 0.832	25.30 / 0.772	18.84 / 0.749	25.20 / 0.826
GSAioIR (5T)	**29.73 / 0.958**	36.65 / 0.976	**31.23** / 0.888	29.48 / 0.886	23.02 / **0.849**	30.02 / **0.911**
Dense-output models: shared-observation evaluation						
InstructIR (7T)	26.71 / 0.942	35.84 / 0.970	31.22 / **0.889**	**29.59 / 0.889**	22.96 / 0.833	29.26 / 0.904
DFPIR (5T)	29.38 / **0.958**	**37.62 / 0.978**	31.20 / 0.887	28.82 / 0.873	**23.85** / 0.842	**30.17** / 0.908

Gaussian adaptations and dense references. The shared-observation comparison groups Gaussian restoration implementations separately from dense-output reference models. GaussianSR-R and GSASR-R are not unmodified published super-resolution models. GSASR-R denotes the reduced-capacity adaptation with approximately 9.264 million parameters; it is not the full published GSASR architecture. Both adaptations are evaluated after 40,000 updates. Their capacity and conditioning differ from GSAioIR. The Avg-3 column covers three restoration families at denoising $\sigma = 25$; it does not imply a common three-task training scope. Full per-image records and inference provenance are retained for each implementation.

For both Gaussian adaptations, paired-task outputs are re-scored from saved predictions. Denoising at $\sigma = 25$ is evaluated again on all 68 shared observations because the earlier noise sequences were not aligned. The shared-observation tables and figures use only these aligned results.

AirNet uses its public three-task model with 8,930,313 parameters and the original modulated deformable-convolution architecture. All 668 common-set predictions are included; the model is not applied to blur or low light, which are outside its training scope.

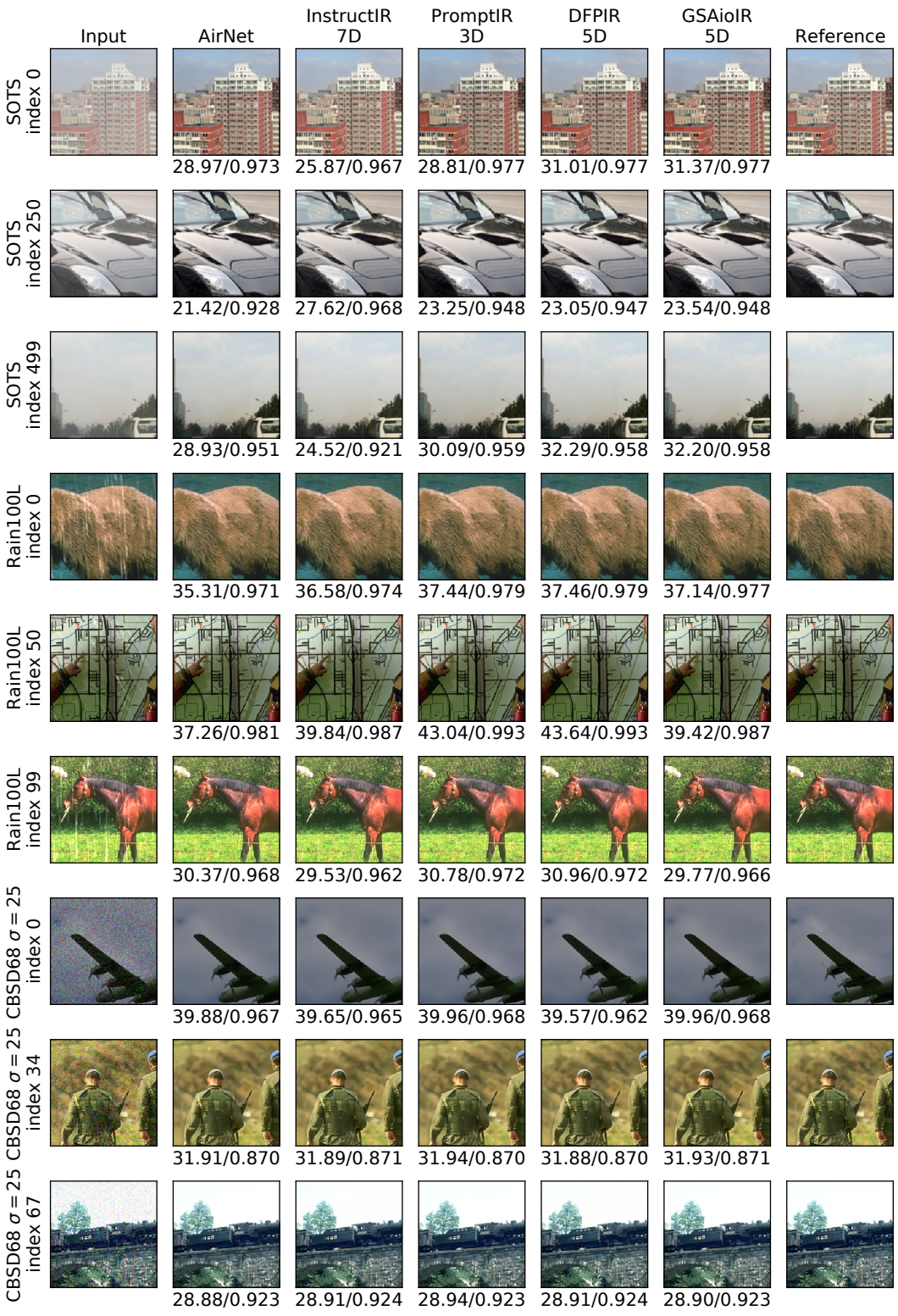

Figure 8: Dense-output comparisons at the first, middle and last indices of SOTS, Rain100L and CBSD68 at $\sigma = 25$. Model scopes and training budgets differ. All numbers are full-image PSNR/SSIM.

Table 16: Paired PSNR analysis on the complete common set. Each task cell is mean GSAioIR-minus-baseline difference (dB) / percentage of images where GSAioIR has higher PSNR. Macro-3 intervals resample SOTS scene-prefix groups and images in the other datasets. They are conditional on fixed models, not training seeds.

Baseline	SOTS	Rain100L	CBSD68	Macro-3 Δ [95% interval]
InstructIR (7D)	$+3.014$ / 73.8%	$+0.816$ / 88.0%	$+0.008$ / 52.9%	$+1.279$ $[+1.106, +1.452]$
PromptIR (3D)	$+0.849$ / 64.4%	-0.764 / 9.0%	-0.053 / 19.1%	$+0.011$ $[-0.093, +0.111]$
DFPIR (5D)	$+0.346$ / 59.0%	-0.966 / 4.0%	$+0.024$ / 52.9%	-0.199 $[-0.301, -0.098]$

Tables 14 and 15 consolidate the shared-observation results. Table 16 adds paired uncertainty estimates rather than repeating the dataset means.

B.3 Fixed qualitative comparisons

Figure 9 uses the lexicographically first sample from each archived Task5 dataset. Selection does not depend on image score. Each column uses the same 160×160 centre crop within a row. The numbers beneath model outputs are their full-image PSNR/SSIM, not crop scores. Scores are computed from the displayed predictions using the same RGB uint8 metric protocol.

GSAioIR uses the released five-task model; GaussianSR-R and GSASR-R use their 40k-update five-task adaptation models. The noisy CBSD68 input is replayed with the original loader and seed 93404. The separately corrupted GSAioIR global view remains part of the historical protocol. For SOTS, source JPEGs are decoded with the metric pipeline's OpenCV reader before figure export. No restoration output is retouched or synthesised.

The following grids use fixed indices selected independently of scores. SOTS and Rain100L include indices 0, 1, 2, $\lfloor n/2 \rfloor$ and $n - 1$; the other datasets retain first, middle and last indices. The extra SOTS/Rain100L rows use earlier archived forwards and are re-scored from the displayed output files. Their small numerical differences from the complete-cohort records do not alter the aggregate tables. Every row displays the same 160×160 centre region across input, methods, and reference. The numbers are full-image PSNR/SSIM, not crop measurements. Models retain their actual training-scope labels. Only supported methods appear in the GoPro and LOLv1 comparisons. All images are decoded and displayed without retouching; metric tables include every test image regardless of its quality.

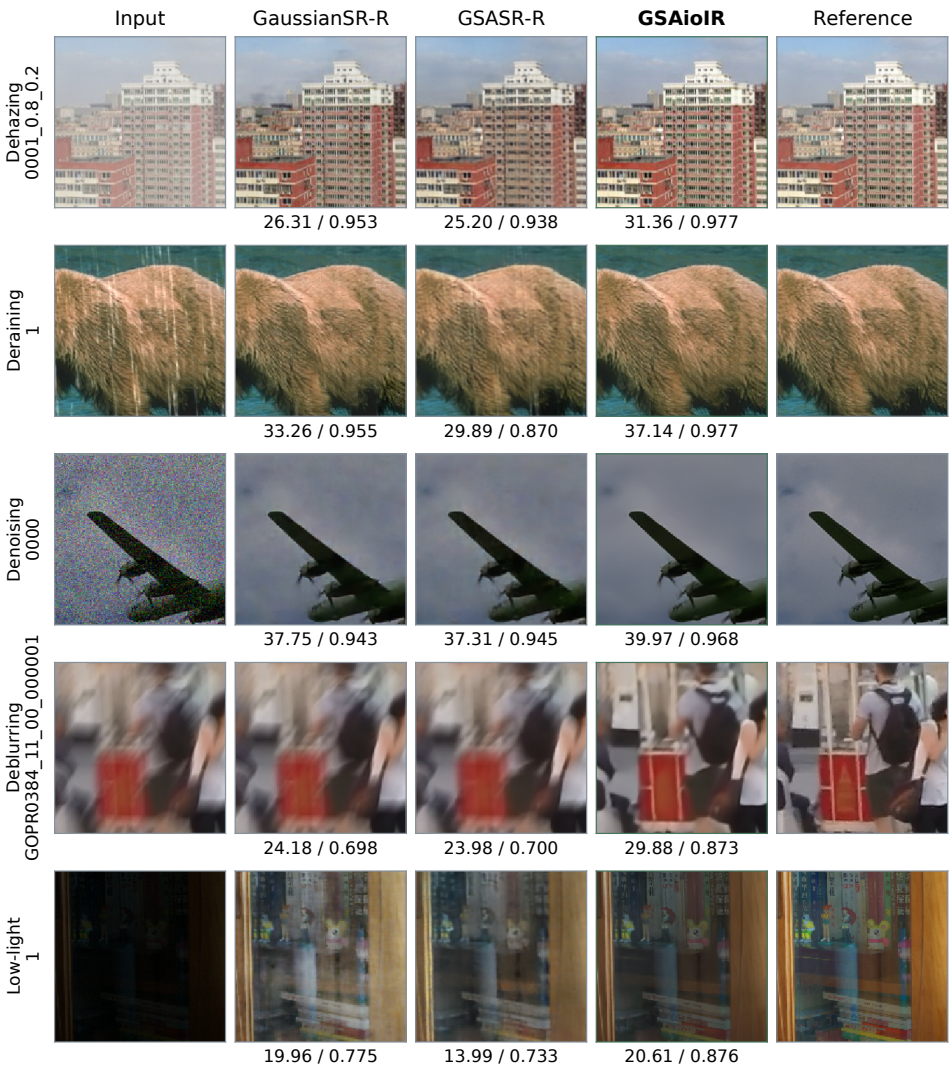

Figure 9: **Five-task qualitative examples.** Identical centre crops from the first archived image in each dataset. Values are full-image PSNR (dB)/SSIM. Input, reference, and three restored outputs are aligned. Training budgets, model capacities, and global/task conditioning differ; these examples do not isolate an architectural advantage.

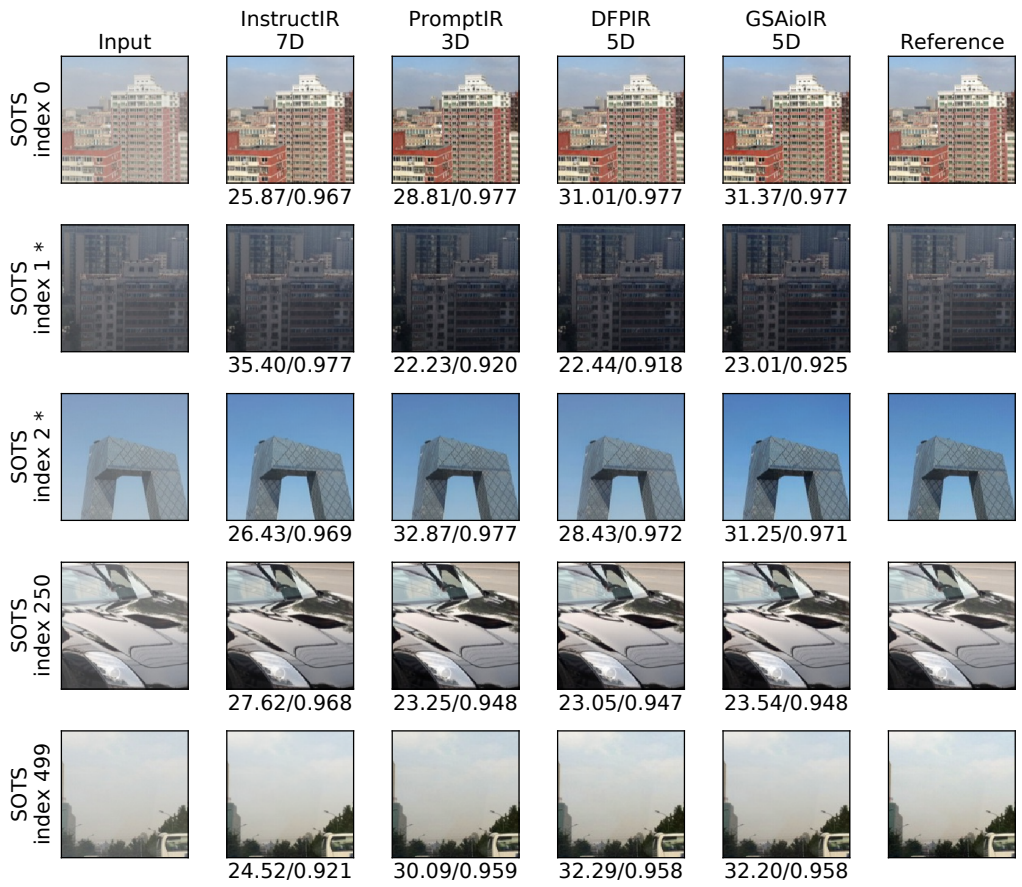

Figure 10: SOTS: indices 0, 1, 2, 250 and 499. Starred rows use additional earlier archived predictions, independently re-scored against the same targets; the full-set aggregate results are unchanged.

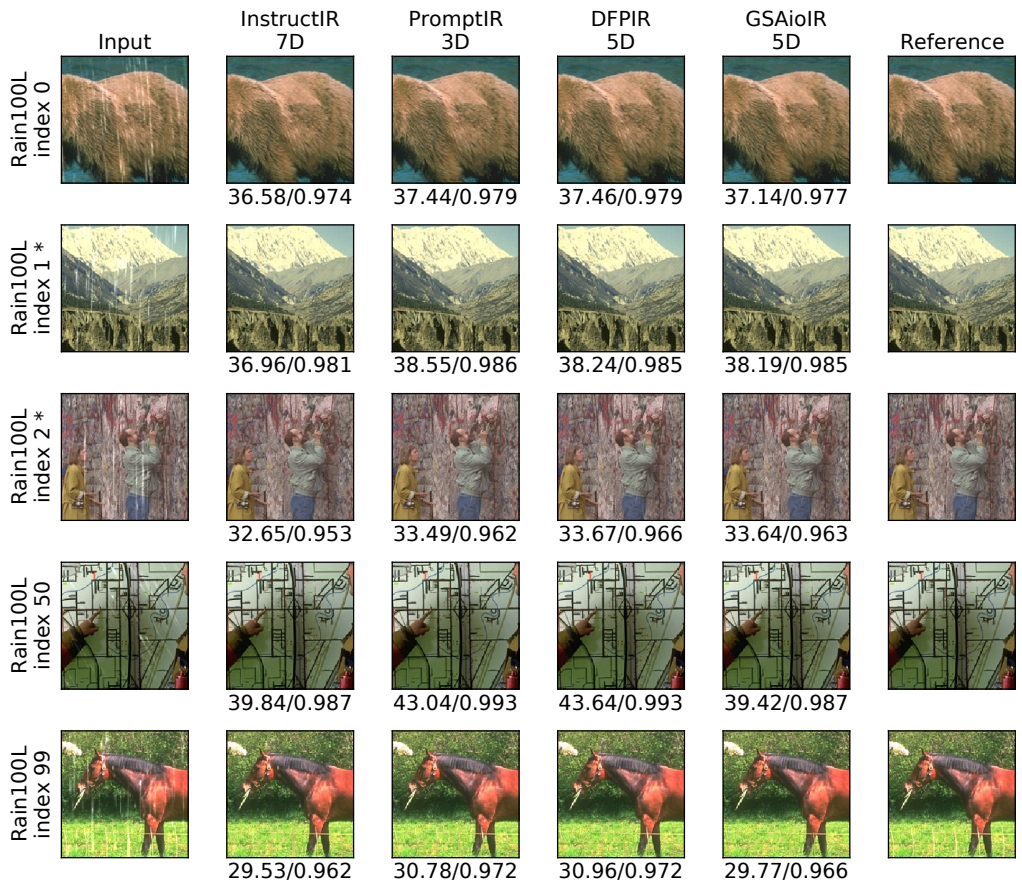

Figure 11: Rain100L: indices 0, 1, 2, 50 and 99. Starred rows use additional earlier archived predictions, independently re-scored. All methods share the same crop coordinates within each row.

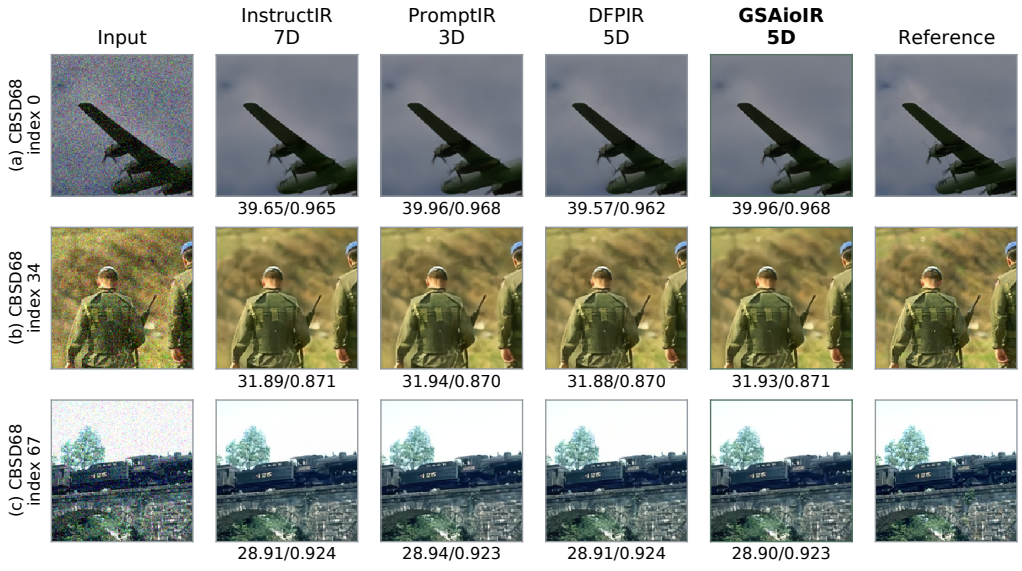

Figure 12: CBSD68 at $\sigma = 25$: fixed entries from the 68-image set. Every model receives the same noisy FP32 observation; GSAioIR context is derived from that observation.

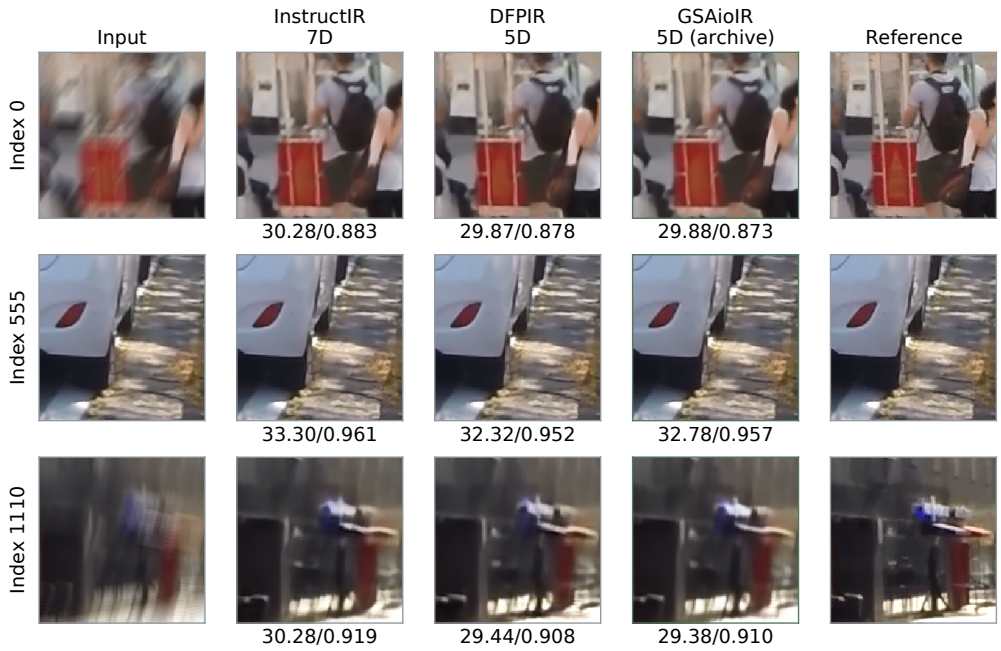

Figure 13: GoPro: fixed entries from the complete 1,111-image set. InstructIR and DFPIR use new full-image predictions. GSAioIR panels match the archived outputs independently re-scored in Table 15.

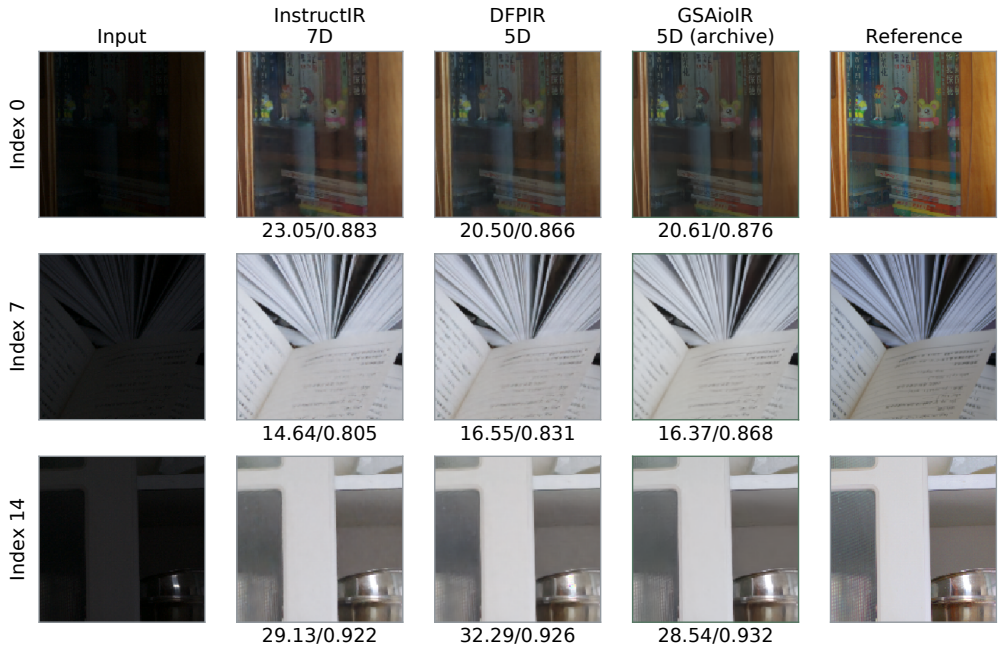

Figure 14: LOLv1: fixed entries from the complete 15-image set. The public PromptIR-3D model is not applied to this unsupported task. GSAioIR panels use the independently re-scored archive.

C Component analysis and controlled diagnostics

C.1 Trained ablations and Gaussian-field behaviour

The eight Task3 ablation runs use the Base model with eight fine slots, seed 3402, batch size 32, and 160×160 crops. Each runs for 25 epochs (111,275 updates). AdamW uses a five-epoch warmup from 2×10^{-4} to 3×10^{-4} and cosine decay to 3×10^{-5}. All eight runs are evaluated at the final epoch. They share the data, optimiser, loss, seed and schedule, with only the designated component changed. Each variant is trained once with the same seed. The comparison measures component effects within this training setting, without estimating variation across independent training runs.

The no-global control bypasses global images, masks, crop boxes, and global attention. It freezes the bypassed modules while retaining their state dictionaries. Its total parameter count therefore cannot be interpreted as its active trainable capacity. The single-field control deletes the coarse and middle heads and predicts sigmoid RGB through the fine branch, retaining guidance and the fine encoder skip. Progressive conditioning introduces learned projections of rendered coarse content into the middle head and of coarse-plus-middle content into the fine head. Gradients propagate through these cross-field inputs.

The width control changes both coarse and middle heads from 256 to 128 channels. The slot control changes their primitives per cell from four to one; the fine branch retains eight slots. Its total primitive count falls from 20,800 to 14,800 per 160×160 crop. Thus this control tests the coarse/middle allocation, not a reduction in fine slots.

Table 17: **Ablations by CBSD68 noise level.** PSNR/SSIM at the end of training under the shared protocol in Appendix C.1. Table 5 reports the other tasks and five-setting mean. C/M denotes coarse/middle. Bold marks the highest PSNR.

Noise	Base model	No global context	No CGW	No fine skip	Single RGB field	Progressive fields	Head width 128	One C/M slot
$\sigma = 15$	**33.88**/.933	33.81/.932	33.85/.933	33.86/.933	33.76/.932	33.85/.933	33.86/.932	33.84/.932
$\sigma = 25$	**31.27**/.889	31.20/.888	31.25/.889	31.24/.889	31.19/.888	31.25/.889	31.25/.888	31.23/.888
$\sigma = 50$	**28.06**/.800	28.01/.798	28.05/.800	28.03/.799	28.01/.798	28.05/.800	28.04/.799	28.03/.799

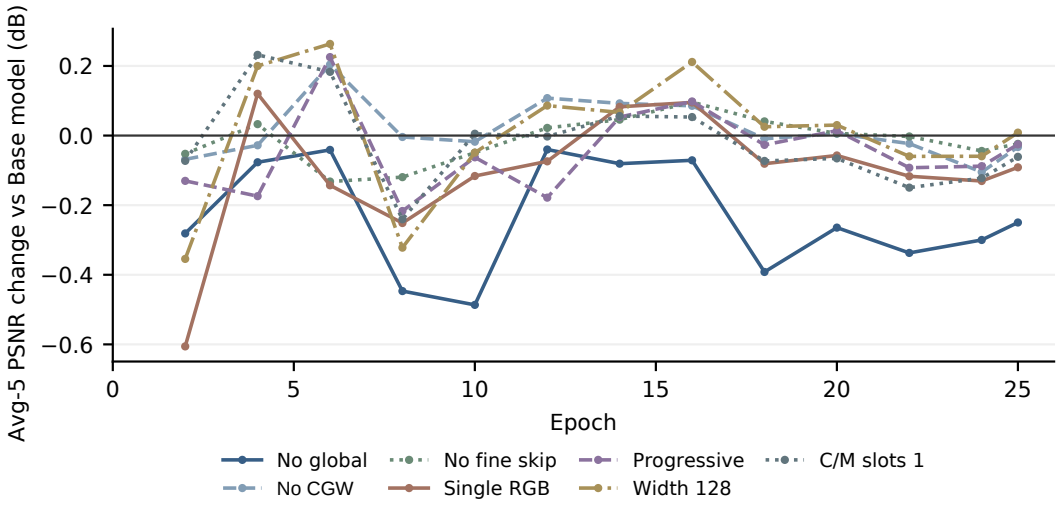

Figure 15: Validation PSNR differences from Base model at matched recorded updates. Unsmoothed curves show how component differences evolve during training. Table 5 compares all variants at the final epoch.

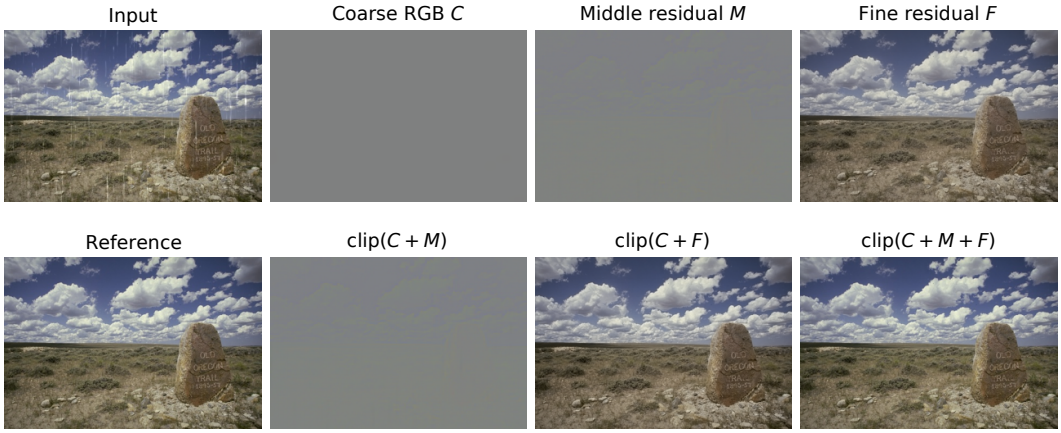

Figure 16: **Field composition on Rain100L image 38.** Coarse RGB C, signed residuals M, F, and fixed-weight field removal at full resolution. Both residuals use the same display range $[-0.55, 0.55]$, mapped to $[0, 1]$. RGB panels use identical clipping. This is one image, not a retrained ablation.

Retrained examples. Figure 6 shows the first stored SOTS and Rain100L examples, with inputs and targets above the four ablation variants. Figure 17 retains all three fixed first-index examples from SOTS, Rain100L, and CBSD68, without selection by score. Each model receives the same FP32 observation and native centre crop. Models with a global branch derive context from that observation. The models are the final Base, no-global, no-CGW, and single-field models. These examples show local effects that can differ from the validation means. Unlike the original denoising validation, they use a single observation for both local input and global context.

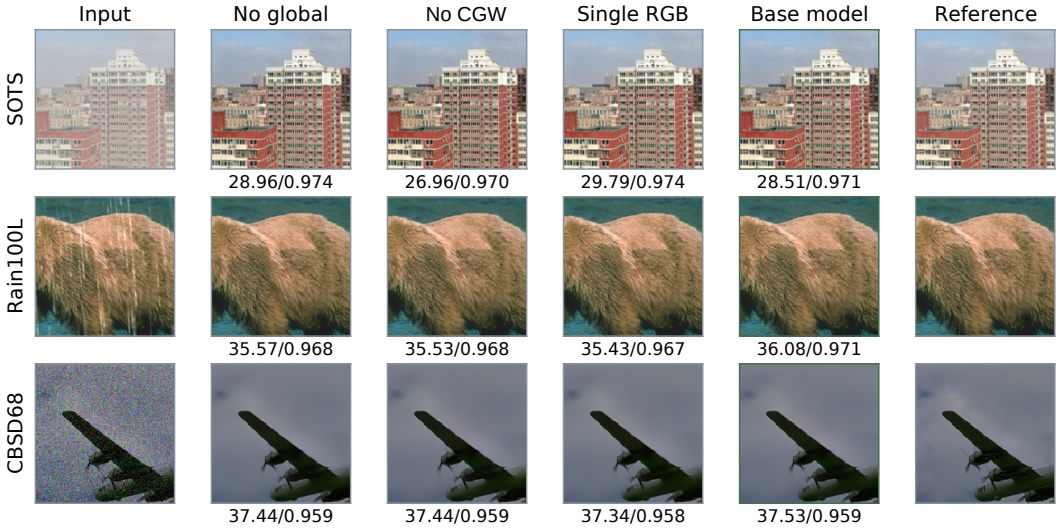

Figure 17: **Complete retrained component examples.** Fixed first-index 160×160 centre crops. Numbers are crop PSNR/SSIM for final 25-epoch models receiving the same observed input. The Base model is not best on the shown SOTS crop. These examples do not replace full validation or reproduce its historical denoising observation protocol.

Table 18: Rain100L image 38: fixed-weight field removal. The full image is 480×320; PSNR is computed against its reference.

Composition	PSNR (dB)	MAE to full output
Degraded input	31.23	0.00814
C	14.93	0.15068
$C + M$	14.94	0.15080
$C + F$	36.15	0.00785
$C + M + F$	**38.75**	0.00000

Fixed-weight interventions. The archived full-resolution forward stores the unquantised coarse RGB, middle residual, fine residual, and final image. We recompute the compositions in Table 18 directly from these tensors, without retraining or rerunning the network. PSNR uses the same clean target, RGB uint8 rounding after clipping, and no border crop. The mean absolute error (MAE) column compares clipped floating-point compositions to the stored full output. It is a consistency diagnostic, not an additional dataset score.

The middle and fine residual ranges are $[-0.04651, 0.06906]$ and $[-0.54255, 0.48530]$, respectively. Their root-mean-square amplitudes are 0.01038 and 0.17807. Figure 16 uses a shared residual display range, so their contrast difference is not created by independent auto-scaling. The near-constant coarse field supplies an RGB offset in this example. These observations concern one image and cannot establish a unique or dataset-wide division of information among the fields.

Complete-cohort analysis. Each of the 668 common-set images contributes its native 160×160 centre crop. The global context uses the same full observed image, and crop-box coordinates identify the crop within it. The released five-task model predicts 27,200 primitives per crop with 12 fine slots. This is the expanded model, distinct from the eight-slot Base model retraining study.

One forward pass stores C, M, and F. The intervention compares $\mathrm{clip}(C + M + F)$, $\mathrm{clip}(C + F)$, $\mathrm{clip}(C + M)$, and $\mathrm{clip}(C)$ against the same crop reference. Clipping is applied after composition. The decomposition and stored-prediction consistency errors are zero on every crop. Field-removal intervals use 10,000 bootstrap replicates with seed 3402, resampling SOTS scene-prefix groups and other datasets' images. The weights, crop policy, and model remain fixed.

Table 19: **Fixed-weight field removal on 668 centre crops.** PSNR in dB; Δ is removal minus full prediction. Brackets give 95% bootstrap intervals; CBSD68 uses $\sigma = 25$.

Dataset (n)	Full	Mean ΔM [95% CI]	Med. ΔM	M improved	Mean ΔF
SOTS (500)	29.27	-0.26 $[-0.39, -0.13]$	$+0.22$	277/500	-17.26
Rain100L (100)	35.89	-2.33 $[-2.71, -1.99]$	-1.79	2/100	-23.02
CBSD68 (68)	30.59	-0.53 $[-0.66, -0.43]$	-0.37	0/68	-17.93

Figure 4 in the main text displays the complete-cohort distributions.

The mean spatial standard deviation of C is 0.00246 on SOTS, 0.00026 on Rain100L, and 0.00042 on CBSD68. Corresponding values for F are 0.21281, 0.18813, and 0.20323. Fine residuals thus contain much more spatial variation in these evaluated crops. That allocation does not establish a general semantic role for each field. In particular, the negative mean and positive median of the SOTS middle-field intervention indicate heterogeneous effects. A fixed-weight removal tests a learned decomposition, whereas a retrained single-field model can redistribute information.

C.2 Observation protocol and computational cost

We execute both released models on all 68 CBSD68 images at each of three noise levels. The paired modes share the exact local floating-point noisy tensor. Historical mode constructs a bounded view from the clean target and adds independent noise. Single-observation mode constructs its bounded view directly from the local noisy tensor, with no additional noise or intermediate uint8 quantisation. Ground truth is used only for scoring in the latter mode.

Seeds are 93403, 93404, and 93405 for $\sigma = 15, 25, 50$, reset once per complete dataset. Both modes use 160-pixel tiles, overlap 32, batch size one, and the same model, precision, RGB uint8 metric conversion, and zero border crop. A total of 816 predictions covers 408 matched image pairs. The

historical replay reproduces the corresponding visual-archive means within 0.000018 dB PSNR and 0.000004 SSIM, supporting the consistency of the replay.

Table 20: Full CBSD68 paired observation control, 68 images per row. PSNR/SSIM are dataset means; Δ is single-observation minus historical PSNR. Extra precision makes the small measured differences visible.

Model	σ	Historical	Single observation	Δ (dB)
Task3	15	34.08563/0.935390	34.08551/0.935386	-0.000116
Task3	25	31.45957/0.892910	31.45947/0.892904	-0.000096
Task3	50	28.23710/0.806654	28.23726/0.806589	$+0.000153$
Task5	15	33.83513/0.932526	33.83516/0.932522	$+0.000029$
Task5	25	31.22906/0.887589	31.22894/0.887582	-0.000116
Task5	50	28.03351/0.797773	28.03357/0.797722	$+0.000067$

Table 20 shows small mean differences between the two observation constructions for these fixed models. It therefore suggests limited sensitivity in this CBSD68 evaluation. The Task3 visual model cannot be independently linked to the selected update in the original result summary, so this control is not a replay of that table. Task5 results at $\sigma = 15, 50$ are additional diagnostics and do not enter its five-task average. These observations do not establish that the two protocols are interchangeable during training or for other models.

Table 21: Controlled inference profiling on one RTX 4090, FP32, batch one. Latency is mean $\pm$ standard deviation over ten runs after five warmups. Memory is peak PyTorch allocation at 720×1280. Image-network parameters exclude external text encoders; preprocessing and I/O are excluded.

Model	Params (M)	256^2 (ms)	512^2 (ms)	720×1280 (ms)	Memory (MiB)
InstructIR-7D	15.84	$\mathbf{8.59} \pm 0.01$	$\mathbf{21.15} \pm 0.02$	$\mathbf{109.34} \pm 0.02$	1098.0
PromptIR-3D	35.59	55.33 ± 0.07	233.81 ± 0.03	867.71 ± 0.11	8198.5
DFPIR-5D	31.10	55.54 ± 0.08	235.51 ± 0.06	876.70 ± 0.09	8386.3
GSAioIR-5D	27.85	85.84 ± 0.21	331.63 ± 0.80	1245.93 ± 1.27	**266.7**

Inference profiling. Profiling runs one model at a time on the same RTX 4090 with no competing GPU process. All environments use PyTorch 2.3.1 and CUDA 12.1. TensorFloat-32 is disabled, cuDNN benchmarking is disabled, and CPU thread counts are fixed to four intra-operation and one inter-operation thread. One fixed degraded image is resized solely to construct the three profiling shapes; these resized inputs are not used for accuracy measurements.

Models use FP32 and batch size one. Each shape has five warmup forwards and ten measured forwards, with CUDA synchronisation before and after wall-clock timing. Peak allocated memory is reset after warmup and includes the active evaluation wrapper's GPU tensors. It is not a measurement of all driver-reserved device memory. Native full-image prediction is used for the three baselines; GSAioIR retains 160-pixel tiles, overlap 32, and tile batch size one. All twelve profiles finish without an OOM fallback.

Input tensors and GSAioIR's bounded global input are already resident on the GPU. Data loading, context resizing, and text encoding are excluded. Fixed task/text embeddings are prepared before timing. GSAioIR clears its global-feature cache before every forward, so each measurement recomputes global features once and reuses them only across tiles of that image. Parameters in Table 21 count the restoration network, not external text encoders. Per-repeat timings, image sizes, parameter counts, device identity, and backend snapshots are retained. The table characterises this controlled setup rather than latency variability across all images or deployment hardware.

Remaining controls. The component study examines the current Gaussian design. A matched dense RGB head and a separate crop-geometry control would help distinguish the output representation from the conditioning strategy. Repeated training runs would assess the variability of the observed differences, while end-to-end timing would include preprocessing and task encoding.

The downstream experiments in Appendix D.1 test adaptation to clean-input prediction tasks. Whether restoring degraded images improves a fixed recognition model is a separate question that these experiments do not evaluate.

D DOWNSTREAM TRANSFER

D.1 TASK BRANCHES AND TRAINING PROTOCOL

Gaussian task fields and readouts. The downstream model jointly fine-tunes the restoration feature path and added Gaussian task fields. Segmentation and detection have independent fields; depth and normals share a field with separate readouts. Anisotropic Gaussian aggregation produces features on a 40×40 grid. The detector builds 20×20 and 10×10 levels by downsampling those rendered features, rather than by using three independent Gaussian renderers. Its feature pyramid (Lin et al., 2017a) supports an FCOS-style anchor-free readout (Tian et al., 2019), with centre-sampling radius 1.5 strides and box-distance ranges 0–32, 32–64, and above 64 pixels.

Detection combines focal classification loss (Lin et al., 2017b), centerness-weighted GIoU regression (Rezatofighi et al., 2019), and centerness BCE. Segmentation combines cross-entropy with a soft Dice term (Milletari et al., 2016), weighted by 0.2. Depth uses log-depth L1 and valid-neighbour gradient L1 with weight 0.1; normals use cosine loss and a valid-neighbour gradient term with weight 0.1. Dense readouts use context aggregation and shallow RGB guidance. This is a hybrid of explicit Gaussian feature aggregation and neural readouts, not a purely Gaussian pixel predictor. Original restoration RGB readouts are bypassed; their restoration accuracy after downstream fine-tuning is not evaluated.

Joint training. The restoration-pretrained backbone and task branches are jointly fine-tuned on interleaved CamVid, Penn–Fudan and NYUv2 microbatches. NYUv2 supervises depth and normals together. The final model has 7,240 downstream updates, including a one-hour continuation on four RTX 4090 GPUs. Segmentation and geometry participate in all 7,240 updates, while detection participates in 5,054.

AdamW uses weight decay 10^{-4} and gradient clipping at three. During the final stage, backbone and task-branch learning rates warm for 100 updates to 10^{-5} and 10^{-4}, respectively, then decay to one tenth of those values. Task loss weights are 1, 1, 0.5 and 0.5 for segmentation, detection, depth and normals. Training uses FP16 mixed precision with FP32 losses. Detailed batch schedules are retained in the experiment records.

Independent single-task models. SegNetSmall is a compact SegNet-style implementation, not the original full-size SegNet. Detection uses Faster R-CNN MobileNetV3-320, with the image resize fixed to 160×160. Depth and normals use two independent DPT-Hybrid instances adapted from the Omnidata implementation. Neither instance loads ImageNet or Omnidata pretrained weights. Depth is positive through softplus; normals are signed unit vectors. These results are not reproductions of publicly pretrained Omnidata performance.

Each single-task model starts from random initialisation and receives 900 seconds of cumulative training-loop time on one GPU with batch size eight. The budget excludes validation, model saving and test evaluation. Training uses AdamW with weight decay 10^{-4} and gradient clipping at three. Continuation preserves the optimiser state, warms the learning rate to 10^{-4} over 100 updates and decays it to 10^{-5}. Depth uses mixed precision; the other models use FP32. Table 22 reports the resulting update counts.

D.2 RESULTS AND PREDICTION COMPARISONS

Evaluation and selection. CamVid uses 367 training, 101 validation and 233 test images. Penn–Fudan uses a fixed 120/25/25 split with seed 20260924, not an official benchmark partition. NYUv2 uses 715 training frames, 80 internal validation frames and the official 654-frame test set. The internal split is frame-based; scene-level separation has not been verified. All methods receive the same 912 saved test inputs; depth and normals share NYUv2 inputs but use separate single-task models. Segmentation aggregates one confusion matrix over the full test set. Detection follows the COCO

Table 22: **Single-task training budgets.** Each model uses one GPU and batch size eight. Training time is rounded to the nearest second.

Model	Seconds	Updates	Training images
SegNetSmall	900	77,475	367
Faster R-CNN	900	11,408	120
DPT-Hybrid depth	900	8,728	715
DPT-Hybrid normals	900	10,180	715

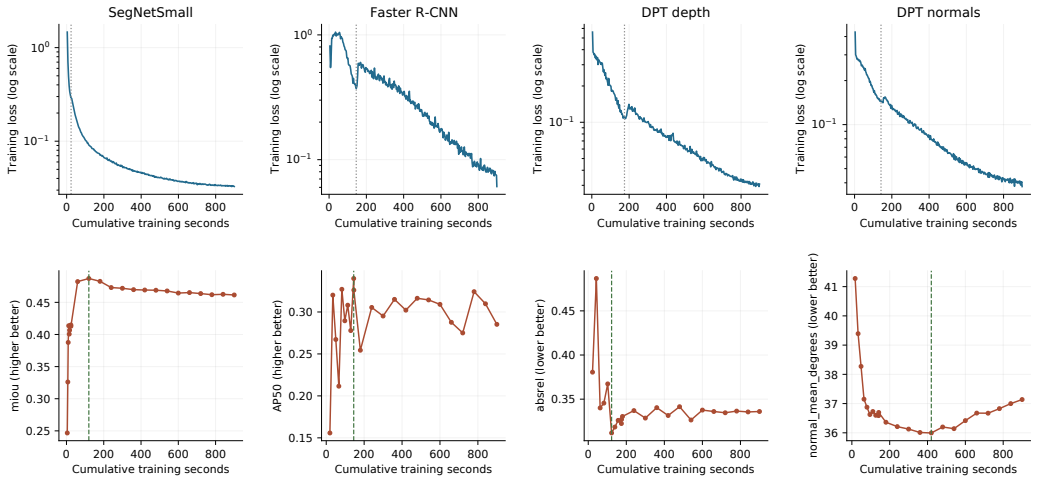

Figure 18: **Single-task training and validation.** Recorded curves for four independent runs with cumulative 900-second budgets. Validation-selected models are reported separately from final models. Curves show the observed optimisation and validation behaviour.

evaluation protocol (Lin et al., 2014); AP averages IoU thresholds from 0.50 to 0.95, whereas AP50 uses 0.50. Geometry metrics aggregate valid pixels.

Table 23 reports the final trained models as the primary results. The separate validation-selected row uses mIoU, AP50, AbsRel and mean normal angle to select the four task-specific models. Selection uses validation scores only; the primary comparison uses the models at the end of training.

Table 23: **Complete downstream results.** Single-task final models receive 900 seconds each; validation-selected models follow their task-specific selection rules. GSAioIR uses the final joint model. Training compute and initialisation differ.

	CamVid		Penn–Fudan			NYUv2 depth			NYUv2 normals	
Setting	mIoU ↑	Pixel acc. ↑	AP ↑	AP50 ↑	AR100 ↑	AbsRel ↓	RMSE (m) ↓	δ_1 ↑	Mean angle (°) ↓	Within 11.25° ↑
Single-task final	0.4034	0.8024	0.0882	0.3262	0.1478	0.2959	1.0296	0.5330	35.9427	0.1897
Validation-selected	0.4117	0.8086	0.1030	0.3588	0.1597	0.2811	1.1005	0.4879	34.9280	0.1902
GSAioIR final	0.4658	0.8461	0.4252	0.8802	0.5164	0.2702	0.9350	0.5598	31.2288	0.2300

Comparison scope and training behaviour. Table 23 shows that the joint model achieves higher mIoU and AP50, lower depth AbsRel and lower mean normal error than the independently trained references. These results support adapting the restoration feature path to semantic and geometric

prediction. The joint model receives restoration pretraining, earlier downstream training and a larger compute budget, so the differences do not isolate the contribution of Gaussian aggregation.

Each configuration is evaluated in one training run; variability across training seeds is not measured. The test partitions were inspected during model development, so the results describe the present experimental setting rather than a new blind evaluation.

Training curves in Figure 18 show that lower training loss need not coincide with better validation performance. The validation-selected references improve some test metrics over their final models, but not all: selecting depth by AbsRel does not also minimise RMSE. Reporting both makes the selection criterion explicit without combining the best test scores from different training stages.

Head-only reference and prediction display. The head-only column uses the archived 1,200-update model with the entire restoration backbone frozen. Only the added Gaussian task fields and neural readouts are trained, totalling 5,559,077 parameters. It uses the same inputs and targets for the displayed examples. The other columns use the final 900-second single-task models and the final 7,240-update joint model. The head-only reference uses earlier readout designs and a shorter budget, so these examples compare available trained systems rather than isolate the effect of freezing the backbone.

Figures 19 and 20 use the first, middle and last test indices for each dataset. Detection shows every final box with score at least 0.5 after NMS at IoU 0.5, with no manual box removal. Segmentation uses one class-ID palette and a shared ignore-255 mask. Depth uses a common 0–10 m display range; raw depths used for evaluation are not clipped to that range. Normal components map from $[-1, 1]$ to RGB using $(n + 1)/2$, with common valid masks.

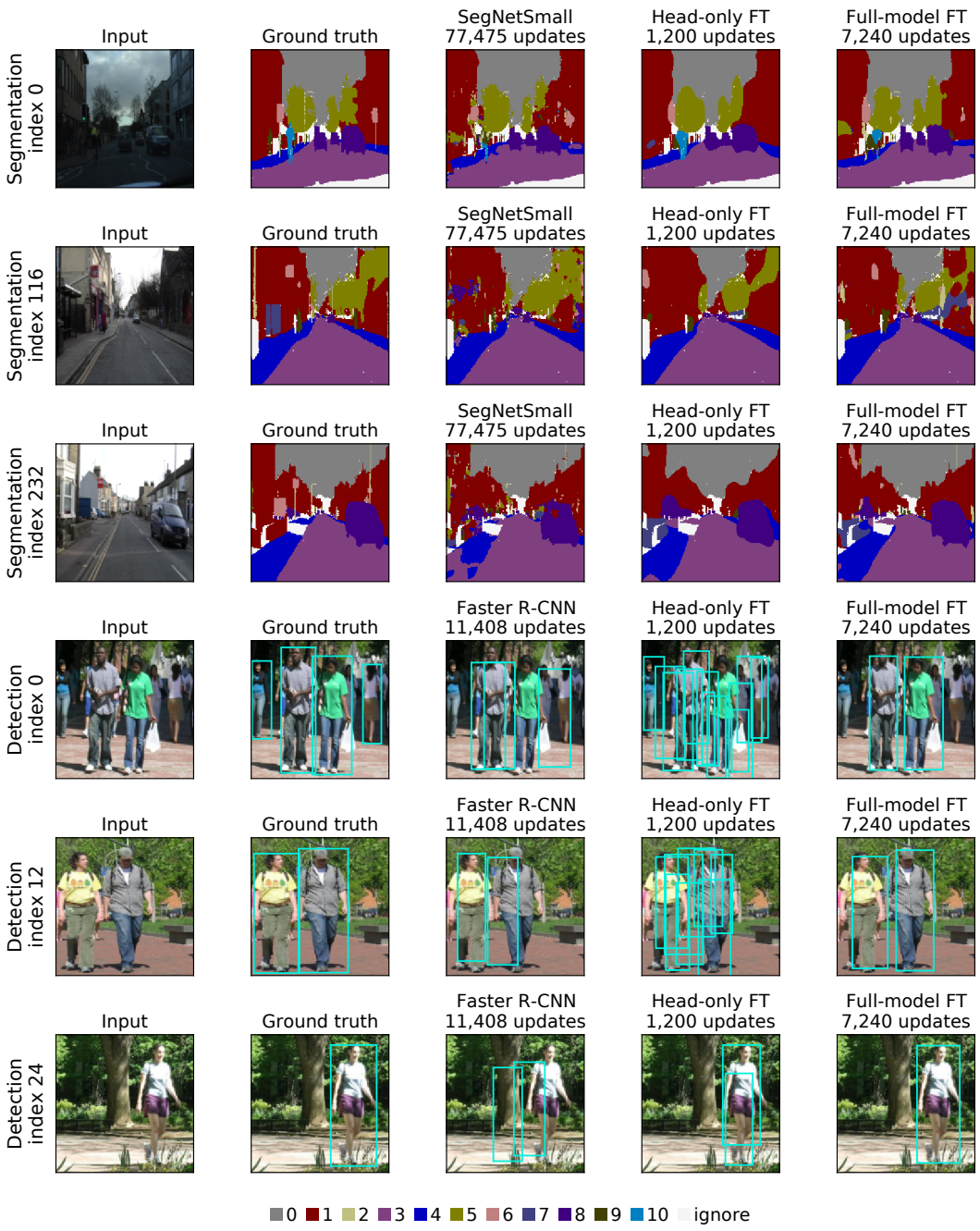

Figure 19: **Fixed segmentation and detection comparisons.** Input, ground truth, task-specific model, head-only fine-tuning, and full-model fine-tuning. First, middle and last test examples share the same display rules; training budgets and readout designs differ.

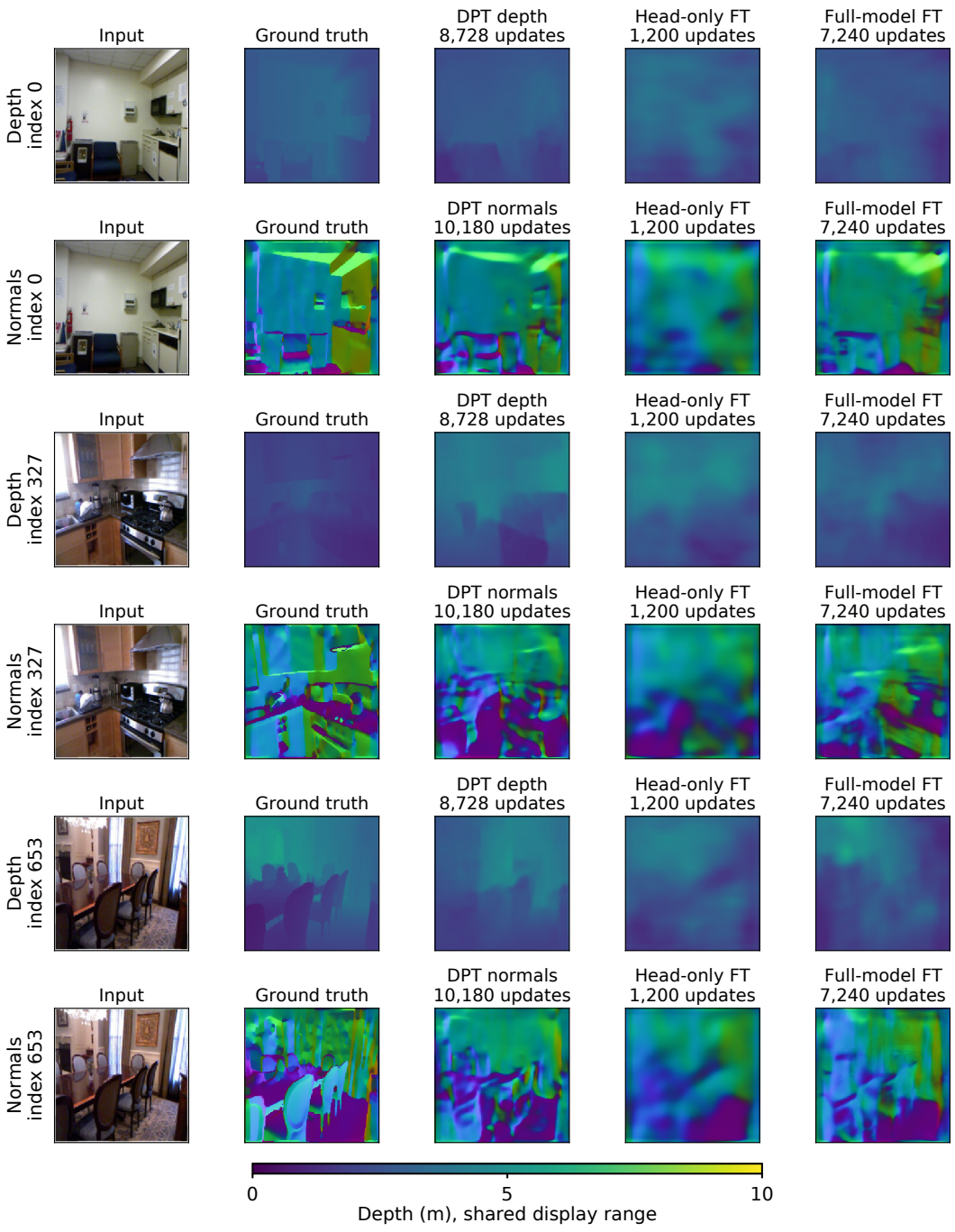

Figure 20: **Fixed depth and normal comparisons.** The same five-column order is used. DPT depth and normal models are trained independently. All methods share the displayed input, valid mask and colour mapping; training budgets and readout designs differ.

Additional semantic examples in Figure 21 use indices $\lfloor 0.2n \rfloor$, $\lfloor 0.4n \rfloor$ and $\lfloor 0.8n \rfloor$, fixed before rendering and disjoint from the first/middle/last set.

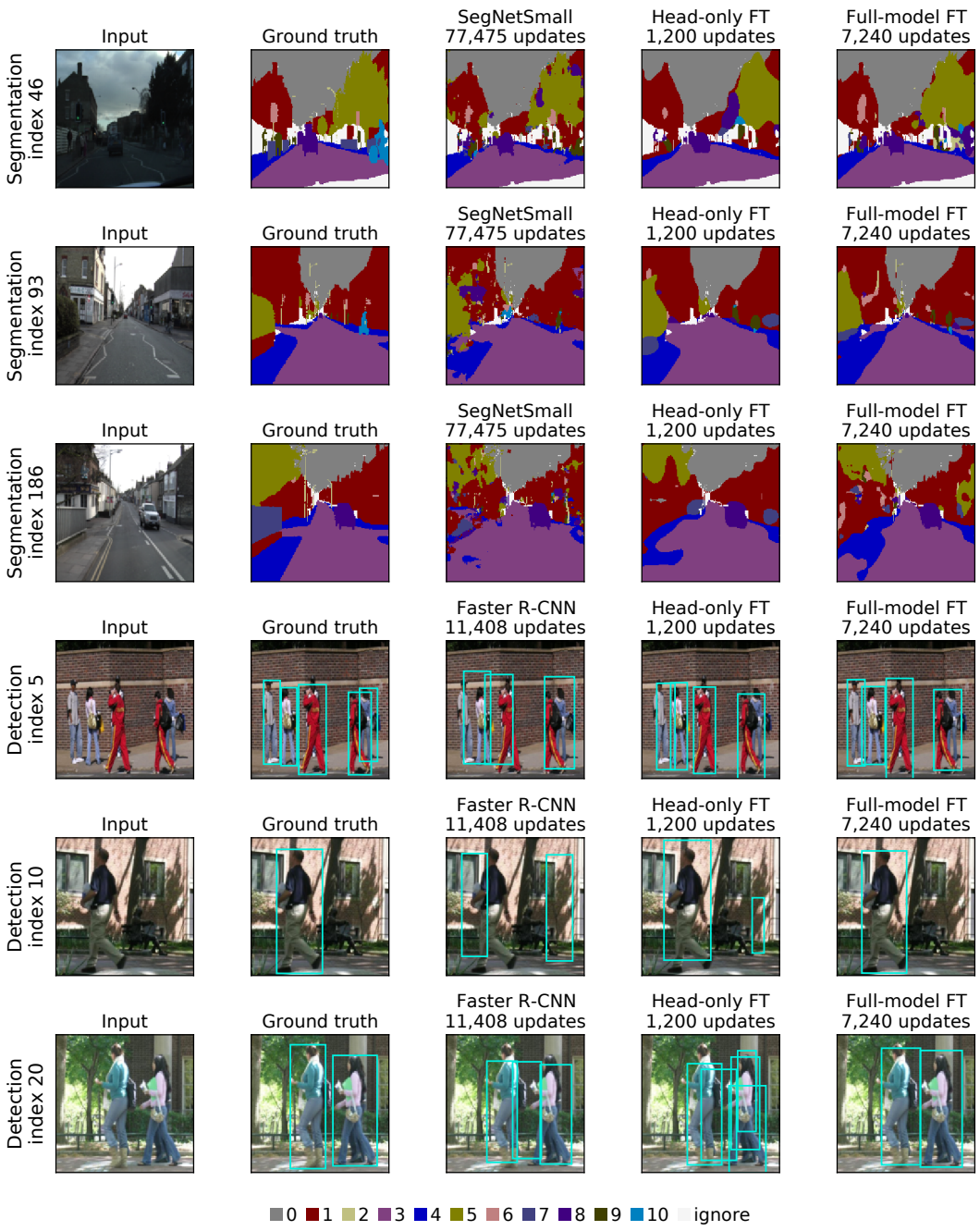

Figure 21: **Additional segmentation and detection examples.** CamVid indices 46, 93 and 186; Penn–Fudan indices 5, 10 and 20. The five columns, weights and display rules follow Figure 19; all retained boxes are shown.