SEGDEM: SEGMENTATION HELPS DEMOSAICING

Anonymous authors
Paper under double-blind review

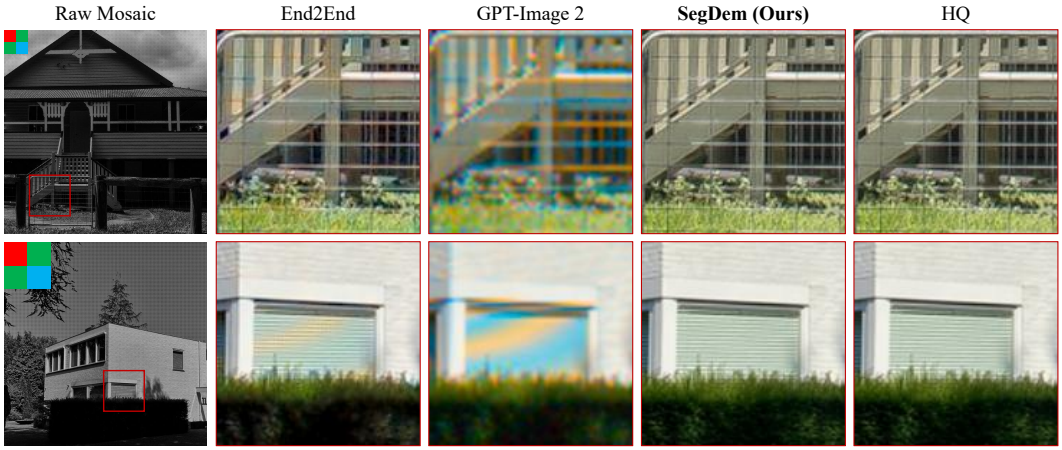


Figure 1: **Motivation of SegDem.** Conventional regression is prone to false colors and moiré artifacts in ambiguous regions, while generative models may hallucinate details unsupported by RAW measurements. SegDem instead transfers structural decoder representations from segmentation to demosaicing, improving boundary-aware reconstruction.

ABSTRACT

Image demosaicing reconstructs a full-color image from incomplete color measurements produced by a sensor covered with a color filter array (CFA). Most existing methods formulate demosaicing as pixel-level reconstruction and mainly rely on local textures, cross-channel correlations, and low-level image statistics. Our core insight is that reconstruction and visual understanding can be viewed as complementary views of shared scene structure: both are grounded in the same underlying physical world, and therefore the structural and physical information inferred from an image should remain consistent across the two tasks. We instantiate this idea with instance segmentation and propose *SegDem*, a cross-task decoder representation transfer framework for demosaicing. SegDem first learns region- and boundary-aware representations through instance-aware structural pretraining and then transfers the decoder to RAW-conditioned reconstruction. Segmentation- and demosaicing-conditioned features are further anchored to a shared frozen DINOv2 representation space to preserve structural organization across tasks. We instantiate SegDem with convolutional, Transformer-based, and state-space backbones for unified Single- and Quad-Bayer demosaicing. Extensive experiments on synthetic, external, and challenging datasets demonstrate improvements across different architectures and CFA layouts.

1 INTRODUCTION

Image demosaicing reconstructs a full-color linear RGB image from spatially incomplete measurements acquired through a color filter array (CFA) (Bayer, 1976; Zeng et al., 2024a). Learning-based methods typically perform end-to-end regression from CFA observations to RGB images (Malvar et al., 2004; Gharbi et al., 2016; Kokkinos & Lefkimmiatis, 2018; Tedla et al., 2025b;a; Zhou et al., 2025b). Although effective, pixel-level reconstruction remains ambiguous around object boundaries,

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.

thin structures, and repetitive patterns, where multiple local color continuations may be consistent with the same CFA observation. (Gharbi et al., 2016; Menon & Calvagno, 2011) Consequently, existing methods can still produce false colors, zipper artifacts, and moiré patterns, particularly when handling CFA layouts with different sampling geometries, such as Single-Bayer and Quad-Bayer patterns (Malvar et al., 2004; Kokkinos & Lefkimmiatis, 2018; Lee et al., 2023; Tedla et al., 2025b).

Recent large-scale generative models have shown that high-level perceptual priors can improve low-level restoration tasks (Wang et al., 2023; Lin et al., 2023; Wu et al., 2024b; Yu et al., 2024; Yang et al., 2026; Sun et al., 2026; Zeng et al., 2025b). However, directly applying generative synthesis to demosaicing is problematic because the reconstructed image should remain anchored to RAW measurements and CFA sampling. Generative restoration may hallucinate textures or shift colors unsupported by the sensor observation (Blau & Michaeli, 2018; Yin et al., 2026; Sun et al., 2026), as illustrated in Figure 1. This motivates us to reconsider the relation between visual understanding and reconstruction. Rather than treating them as isolated objectives, both rely on recovering and representing coherent scene structure: understanding organizes visual content into meaningful regions and boundaries, while faithful reconstruction should recover the structures that make such understanding possible.

From this perspective, high-level understanding can provide structural cues for resolving ambiguous low-level measurements. We use instance segmentation as a concrete instantiation of visual understanding, since it explicitly captures object regions and boundaries (He et al., 2017). Rather than introducing segmentation prediction into demosaicing, we transfer segmentation-pretrained decoder representations to demosaicing, allowing structural knowledge to guide color recovery while preserving RAW measurement fidelity.

Based on this insight, we propose *SegDem*, a cross-task decoder representation transfer framework for unified Single- and Quad-Bayer demosaicing. Stage 1 learns region- and boundary-aware decoder representations through instance-aware supervision, and Stage 2 transfers them to RAW-conditioned reconstruction. To preserve structural consistency across tasks, segmentation- and demosaicing-conditioned decoder features are aligned to a shared frozen DINOv2 representation space (Oquab et al., 2023). Generally, we instantiate SegDem with convolutional (You et al., 2025), Transformer-based (Zamir et al., 2022), and state-space(Guo et al., 2024) backbones.

Our contributions are summarized as follows:

- We introduce a cross-task perspective on demosaicing that connects visual understanding and low level task through shared structural representations.
- We propose SegDem, which transfers region- and boundary-aware decoder representations learned from instance-aware supervision to RAW-conditioned demosaicing without introducing semantic prediction at reconstruction time.
- We introduce DINO-bridged shared-space anchoring, which independently anchors understanding- and reconstruction-conditioned decoder features to a common representation space and helps preserve structural organization during task transfer.
- We validate SegDem with convolutional, Transformer-based, and state-space restoration backbones, demonstrating gains across different architectures and CFA layouts.

2 Related Work

2.1 Image Demosaicing

Demosaicing has long been studied as a reconstruction problem under CFA sampling. Classical methods estimate unobserved channel values with interpolation, edge-aware filtering, and cross-channel correlation priors (Malvar et al., 2004). Learning-based methods replace hand-crafted reconstruction rules with end-to-end networks and often combine demosaicing with denoising to handle realistic RAW degradations (Gharbi et al., 2016; Kokkinos & Lefkimmiatis, 2018). Recent methods further extend this direction to emerging sensor layouts and efficient backbones, including unified Single-, Quad-, and Nona-Bayer demosaicing (Tedla et al., 2025b), multispectral demosaicing with dual cameras (Tedla et al., 2025a), Mamba or Transformer designs for different sensors (Zeng et al., 2024b; Pan et al., 2024; Zhou et al., 2025b;a), and unified architectures for pixel-bin sen-

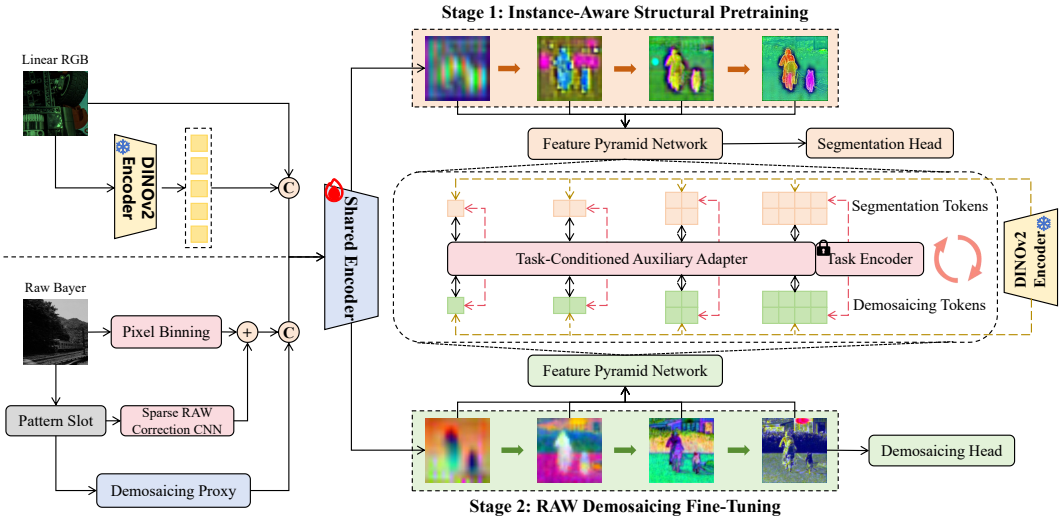


Figure 2: **Overview of SegDem.** Stage 1 learns region- and boundary-aware representations using a clean linear-RGB image state while retaining the mosaicked observation as an auxiliary RAW condition. Stage 2 transfers the pretrained decoder parameters to demosaicing, while task-conditioned auxiliary features and training-only DINOv2 alignment provide structural supervision without altering the sensor-conditioned output pathway.

sors (Lee et al., 2023; Kumar & Yenneti, 2026). Our work differently studies how representations learned through instance-aware structural supervision can be transferred to demosaicing.

2.2 Large-Scale Generative Models for Low-Level Vision

Large-scale generative models have recently become strong priors for low-level vision. Stable-diffusion-based methods use pretrained text-to-image models (Rombach et al., 2022; Labs et al., 2025) for image restoration (Wang et al., 2023; Yang et al., 2023; Lin et al., 2023; Wu et al., 2024b; Yu et al., 2024; Chen et al., 2025; Zeng et al., 2025a), while recent one-step and editing-model-based systems improve efficiency or generalization (Wu et al., 2024a; Yang et al., 2026). These works demonstrate the value of high-level priors, but they also shift restoration toward perceptual synthesis. Recent evaluations report over-generation, semantic inconsistency, and uncertain evaluation practices in generative image restoration (Blau & Michaeli, 2018; Yin et al., 2026; Sun et al., 2026). Our work follows the motivation of using high-level priors, via segmentation guidance.

2.3 Representation Learning

Representation learning transfers high-level visual knowledge without directly generating image content. DINOv2 provides robust object- and region-level features (Oquab et al., 2023), while prior restoration and generative-model studies use vision-language priors or representation alignment to improve downstream visual features (Radford et al., 2021; Luo et al., 2023; Yang et al., 2025; Yu et al., 2025; Singh et al., 2025; Nilaksh et al., 2026; Wang et al., 2026). Related predictive representation methods also learn structure in latent space rather than through direct pixel synthesis (Assran et al., 2023; Mo & Tong, 2024; Grigore & Popa, 2026). SegDem instead uses frozen DINOv2 tokens as a shared training-time reference for segmentation and demosaicing decoder features, while decoder initialization performs the actual cross-task transfer.

3 Method

3.1 Overview

SegDem assumes that visual understanding and reconstruction should preserve consistent structure because they describe the same physical scene. We instantiate understanding with instance segmen-

tation and transfer its learned representation to demosaicing. As shown in Figure 2, Stage 1 uses a clean linear-RGB image state for instance-aware supervision while retaining the mosaicked observation as an auxiliary RAW condition compatible with Stage 2; Stage 2 then transfers the pretrained decoder to linear RGB reconstruction. Decoder initialization performs the cross-task transfer, and task-conditioned features and frozen DINOv2 tokens provide a shared structural reference.

3.2 Instance-Aware Structural Representation Pretraining

Demosaicing recovers a full linear RGB image x from a sparse CFA observation

$$y = \mathcal{M}_p(x) + n, \tag{1}$$

where $\mathcal{M}_p$ is the CFA sampling operator and n denotes sensor noise. Because each sensor location records one color component, reconstruction is under-constrained around boundaries, thin structures, and repetitive patterns. Missing colors should therefore satisfy both CFA measurements and the scene's region and boundary organization. Instance segmentation provides such supervision by explicitly separating object interiors, boundaries, and background. We use it to shape the decoder before transfer rather than to introduce a semantic output during demosaicing.

Stage 1 learns representations rather than a standalone RAW segmenter. The paired clean linear RGB image is used as the image state so that region learning is not dominated by demosaicing errors, while the mosaicked observation remains the RAW condition to preserve the Stage 2 interface. Following Gen2Seg (Khangaonkar & Pirsiavash, 2025), instance masks are converted into dense RGB-coded targets for the same image-to-image decoder family used in restoration. Pixels within an instance are encouraged to share a consistent representation, while adjacent instances and background remain separable. Figure 3 visualizes this supervision; its input is rendered to sRGB only for display.



Figure 3: **Stage 1 segmentation visualization.** From left to right: the sRGB-rendered input, predicted RGB-coded instance mask, and ground-truth mask.

We further use a lightweight instruction adapter to construct task-specific auxiliary views of the shared decoder features. A fixed segmentation prompt and a fixed demosaicing prompt are encoded once by a frozen Qwen2.5-VL (Bai et al., 2025) encoder, producing cached embeddings e_{seg} and e_{demo}. For a decoder feature h_θ^ℓ at level ℓ, the task embedding modulates the normalized feature as

$$\tilde{h}_{\theta,t}^\ell = (1 + \gamma_\ell(e_t)) \odot \text{Norm}\left(h_\theta^\ell\right) + \beta_\ell(e_t), \qquad t \in \{\text{seg}, \text{demo}\}, \tag{2}$$

where γ_ℓ and β_ℓ predict channel-wise scale and shift. In Stage 1, e_{seg} produces instance-oriented auxiliary features for the RGB-coded instance head and representation alignment. The adapter selects only the auxiliary objective without changing the sensor-conditioned output pathway; its architecture and prompts are detailed in Appendix E. The Stage 1 objective is

$$\mathcal{L}_{\text{stage-1}} = \lambda_{\text{inst}}\mathcal{L}_{\text{inst}} + \lambda_{\text{repr}}^{\text{seg}}\mathcal{L}_{\text{repr}}^{\text{seg}}, \tag{3}$$

where $\mathcal{L}_{\text{inst}}$ supervises the RGB-coded instance map and $\mathcal{L}_{\text{repr}}^{\text{seg}}$ anchors the segmentation decoder tokens to frozen DINOv2 tokens. After pretraining, the segmentation head is discarded.

3.3 RAW-Conditioned Demosaicing

Stage 2 combines a proxy linear-RGB state from a frozen lightweight demosaicer with an explicit RAW condition (Figure 2). Because the proxy may mix color evidence across CFA phases, it is only an image-space starting point and is complemented by a sensor branch that retains measured CFA evidence. Pattern-aware packing forms $c_{\text{pack}} \in \mathbb{R}^{4\times H/2\times W/2}$, pixel binning gives $c_{\text{bin}} \in \mathbb{R}^{4\times H/4\times W/4}$, and sparse RAW correction preserves phase-sensitive information before backbone injection. Backbone-specific implementations are detailed in Appendix C.

The reconstruction objective combines linear-domain fidelity, rendered-domain fidelity, perceptual supervision, and edge preservation:

$$\mathcal{L}_{\text{rec}} = \lambda_{\text{lin}}\mathcal{L}_{\text{char}}(\hat{x}, x) + \lambda_{\text{srgb}}\left\|\Gamma(\hat{x}) - \Gamma(x)\right\|_1 + \lambda_1\mathcal{L}_{\text{lpips}}\big(\Gamma(\hat{x}), \Gamma(x)\big) + \lambda_2\mathcal{L}_{\text{edge}}(\hat{x}, x), \tag{4}$$

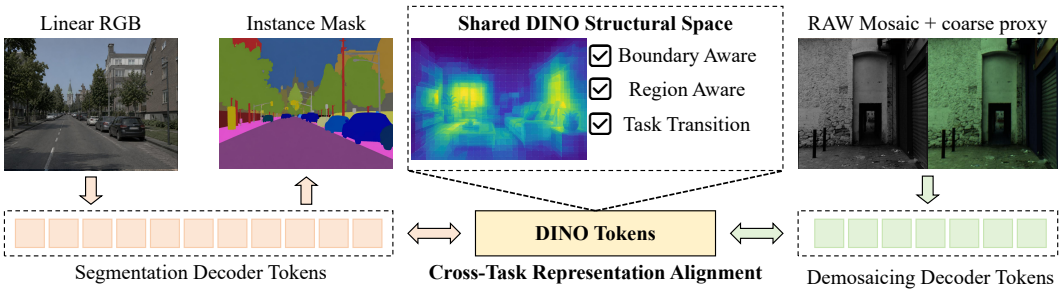


Figure 4: **Motivation for DINO-bridged structural anchoring.** Segmentation- and demosaicing-conditioned decoder tokens are independently aligned to the same frozen DINOv2 representation space, providing a common structural reference without directly matching task-specific features.

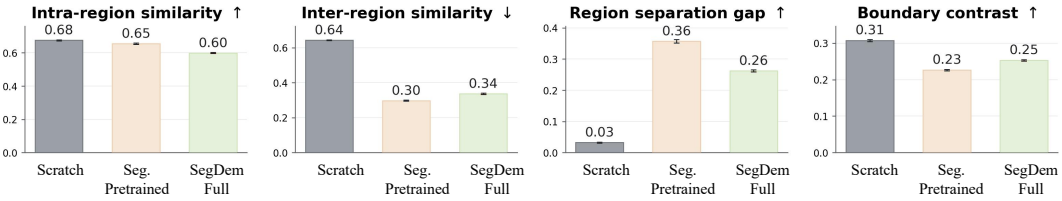


Figure 5: **Decoder representation analysis.** Scratch training yields local boundary responses but limited separation between regions. Structural pretraining produces more region-discriminative decoder features, and the full model largely preserves this organization after demosaicing fine-tuning.

where $\mathcal{L}_{\text{char}}(a,b) = \frac{1}{|\Omega|}\sum_{u\in\Omega}\sqrt{(a_u-b_u)^2+\epsilon_{\text{char}}^2}$ with $\epsilon_{\text{char}} = 10^{-3}$. The fixed PTP operator $\Gamma(\cdot)$ renders camera-linear RGB into sRGB.

3.4 Structural Representation Anchoring and Objective

Decoder initialization transfers the Stage 1 structural representation, but reconstruction fine-tuning may overwrite it with features optimized for local pixel regression. Although segmentation and demosaicing have different objectives, both describe the same underlying scene and should preserve consistent spatial structure. Directly matching their task-specific features is nevertheless undesirable because their feature distributions and optimization objectives differ. We therefore use frozen DINOv2 tokens as a shared structural reference, allowing both task-conditioned representations to be anchored to a common scene representation without directly forcing them to match each other.

Figure 4 illustrates the role of this shared space. In Stage 1, instance supervision introduces region consistency and boundary sensitivity, while $\mathcal{L}_{\text{repr}}^{\text{seg}}$ anchors the resulting segmentation-conditioned tokens to DINOv2. In Stage 2, the transferred decoder is optimized for demosaicing, and $\mathcal{L}_{\text{repr}}^{\text{demo}}$ aligns the demosaicing-conditioned tokens to the same frozen space. Decoder initialization therefore transfers structural knowledge across tasks, while DINOv2 provides a common reference that encourages both representations to preserve consistent scene structure.

For each task t, the multi-scale projection head maps the modulated decoder features to dense tokens as $z_{\theta,t} = \mathcal{P}(\{\tilde{h}_{\theta,t}^{\ell}\}_{\ell})$. Frozen DINOv2 target tokens z_{dino} are extracted from the rendered target $\Gamma(x)$ and reshaped to the same spatial grid. We align tokens at corresponding spatial locations rather than collapsing them into a global feature, because demosaicing errors are strongly location dependent and often concentrate near boundaries or periodic structures. The representation loss is

$$\mathcal{L}_{\text{repr}}^{t} = \frac{1}{|\mathcal{M}|}\sum_{i\in\mathcal{M}} \text{SmoothL1}\left(\text{LN}(z_{\theta,t}^{i}), \text{LN}(z_{\text{dino}}^{i})\right), \qquad t \in \{\text{seg}, \text{demo}\}, \tag{5}$$

where $\text{LN}(\cdot)$ denotes token-wise layer normalization and the full token grid contains $N = 256$ locations. Layer normalization reduces cross-task scale differences, while Smooth L1 preserves task-specific low-level information.

Table 1: **Synthetic Single- and Quad-Bayer demosaicing.** Completed SegDem variants improve over recent baselines; Restormer and MambaIR obtain the strongest fidelity metrics, while EDMUNet gives the best LPIPS. Params(M) reports the full network size.

Method	Params(M)	Single Bayer					Quad Bayer				
		Linear RGB		sRGB			Linear RGB		sRGB		
		PSNR↑	SSIM↑	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	PSNR↑	SSIM↑	LPIPS↓
Jd3Net	3.20	41.1088	0.9798	30.7496	0.8830	0.1573	39.0379	0.9707	29.5206	0.8656	0.2036
BMTNet	12.17	42.2342	0.9823	32.7275	0.9022	0.1334	38.6848	0.9759	30.5340	0.8880	0.1746
FFNet	43.26	44.3537	0.9869	33.6995	0.9178	0.1123	34.4464	0.9211	28.8607	0.8253	0.2267
ms-demosaic	10.68	44.3198	0.9861	33.9864	0.9197	0.1012	42.5651	0.9769	33.0368	0.9159	0.1211
SegDem(E)	78.81	44.5753	0.9881	34.4157	0.9176	**0.0832**	43.1215	0.9858	33.8543	0.9224	**0.0954**
SegDem(R)	34.28	**45.5654**	**0.9908**	**35.1883**	**0.9312**	**0.0891**	**43.9275**	**0.9884**	**34.0871**	**0.9249**	**0.1061**
SegDem(M)	5.45	**45.5766**	**0.9907**	**35.1304**	**0.9327**	0.0951	**43.9385**	**0.9883**	**34.0355**	**0.9261**	0.1129

Table 2: **Real-world RAW evaluation.** No-reference IQA on the rendered sRGB outputs of the real RGGB and Quad-Bayer test sets. ZRR-Huawei is used for Single Bayer, and the MIPI 2022 Quad-Bayer split is used for Quad Bayer.

Method	Real RGGB (ZRR-Huawei)					Real Quad Bayer (MIPI 2022)				
	NIQE↓	MUSIQ↑	MANIQA↑	TOPIQ↑	AFINE-NR↓	NIQE↓	MUSIQ↑	MANIQA↑	TOPIQ↑	AFINE-NR↓
Jd3Net	6.8594	38.0609	0.2621	0.2159	-0.7590	3.7525	50.0181	0.2712	**0.3964**	-0.9019
BMTNet	**6.3338**	42.0402	0.2771	**0.3358**	-0.7295	**3.3781**	50.9180	0.2743	0.3810	-0.9144
FFNet	6.7643	41.9613	0.2613	0.2527	-0.7382	4.0931	49.3128	0.2734	0.3672	-0.8125
ms-demosaic	6.6849	**42.1674**	**0.2797**	0.2994	-0.7024	3.8542	50.8237	0.2694	0.3803	-0.9104
SegDem(E)	**5.7243**	39.6742	0.2367	0.2732	**-0.7613**	**3.3781**	**51.6524**	0.2743	0.3893	**-0.9192**
SegDem(R)	6.4232	**44.8520**	**0.2862**	**0.3416**	**-0.8085**	**3.4242**	**51.9278**	**0.2824**	**0.3991**	**-0.9149**
SegDem(M)	6.5901	39.5615	0.2475	0.3013	-0.7537	3.4450	51.3760	**0.2775**	0.3937	-0.9133

For each image, we randomly sample a subset $\mathcal{M}$ of tokens for alignment. This selective supervision prevents DINOv2 features from over-constraining the decoder and keeps measurement fidelity governed by the reconstruction objective. The full Stage 2 objective is

$$\mathcal{L}_{\text{stage-2}} = \mathcal{L}_{\text{rec}} + \lambda_{\text{repr}}^{\text{demo}} \mathcal{L}_{\text{repr}}^{\text{demo}}. \tag{6}$$

The representation analysis in Figure 5 further explains the effect of the proposed transfer and anchoring. Scratch training can produce strong local responses around visible edges, but features from different regions remain highly similar, indicating that edge activation alone does not yield region-discriminative organization. Instance-aware pretraining reduces between-region similarity and enlarges the separation gap. After demosaicing fine-tuning, the full model retains much of this organization, supporting the view that decoder initialization transfers the structural prior and DINO alignment helps preserve it.

4 Experiments

4.1 Experimental Setup

We evaluate SegDem on unified Single- and Quad-Bayer demosaicing. We use an image-level 7:1:2 split of LSDIR (Li et al., 2023) for demosaicing training, validation, and testing. COCO (Lin et al., 2014) is used exclusively for instance-aware structural pretraining. The LSDIR test split is disjoint from both training stages and checkpoint selection. RAW inputs are synthesized with the corresponding CFA layouts and Gaussian sensor noise, and the ground-truth targets are linear RGB images. We report reconstruction metrics in linear RGB and rendered sRGB, where sRGB evaluation reflects color and boundary errors after the fixed rendering transform $\Gamma(\cdot)$. The evaluated baselines are Jd3Net (Tedla et al., 2025b), BMTNet (Zhou et al., 2025b), FFNet (Zhang et al., 2026), and ms-demosaic (Tedla et al., 2025a). We evaluate the framework with representative convolutional, Transformer, and state-space restoration backbones: EDMUNet (You et al., 2025), Restormer (Zamir et al., 2022), and MambaIR (Guo et al., 2024). In tables, E, R, and M denote the three SegDem instantiations, respectively.

4.2 Evaluation on Synthetic Benchmarks

Table 1 compares SegDem with recent demosaicing baselines. The full SegDem variants improve over the strongest baselines on both CFA layouts, showing that the transferred decoder representa-

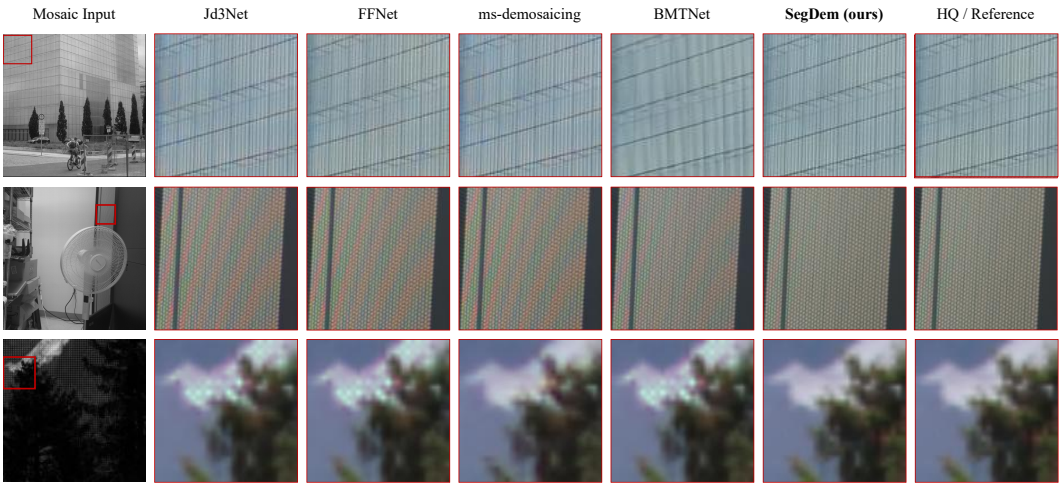


Figure 6: **Qualitative comparison across evaluation settings.** From top to bottom, the three rows show examples from the synthetic data, real captured RAW data, and MIT-Hard, respectively. Reference images are shown when paired ground truth is available; the real captured comparison otherwise reports only rendered predictions.

tion is not specific to one sampling pattern. Restormer and MambaIR give the strongest PSNR/SSIM results, while EDMUNet gives the lowest LPIPS. This difference is consistent with the complementary reconstruction properties of the backbone families: attention and state-space models favor fidelity-oriented reconstruction, whereas the convolutional encoder–decoder is particularly effective at perceptual boundary and texture recovery.

The gain is also more visible after sRGB rendering. Since the fixed rendering transform amplifies chromatic errors and boundary artifacts, improvements in sRGB PSNR/SSIM and LPIPS indicate that the learned structural representation mainly helps in color-ambiguous regions rather than only reducing small linear-domain residuals.

The first row of Figure 6 complements these metrics by highlighting color bleeding near boundaries, zipper artifacts around high-frequency structures, and texture fidelity within coherent regions.

4.3 Evaluation on Real Captured RAW Datasets

We further evaluate on two real RAW datasets that cover the two CFA layouts: the Huawei split of the Zurich RAW-to-RGB (ZRR) dataset (Zhang et al., 2021) for Single Bayer (1204 frames of 448×448, native RGGB) and the quad-Bayer split of the MIPI 2022 Quad-Bayer Re-mosaicing dataset (Yang et al., 2022) for Quad Bayer (70 frames of 1800×1200, native Quad-Bayer). Neither dataset has a clean linear RGB reference, so all methods are rendered to sRGB with the same fixed pipeline $\Gamma(\cdot)$ and assessed with no-reference IQA metrics, namely NIQE, MUSIQ, MANIQA, TOPIQ, and AFINE-NR.

Table 2 shows that SegDem remains competitive on real RAW inputs. On RGGB, SegDem(R) obtains the best MUSIQ, MANIQA, TOPIQ, and AFINE-NR, while SegDem(E) gives the lowest NIQE. On Quad Bayer, SegDem(R) achieves the best MANIQA and TOPIQ with competitive MUSIQ, and SegDem(E) ties for the best NIQE. The second row of Figure 6 further shows cleaner boundaries and fewer structured color artifacts on captured RAW inputs.

4.4 Evaluation on the MIT-Hard Dataset

We further evaluate on MIT-Hard (Gharbi et al., 2016), whose high-frequency patterns amplify CFA ambiguity and are particularly prone to moiré and false-color artifacts.

Table 3 shows that Quad Bayer is consistently more challenging than RGGB because repeated local sampling groups make fine periodic structures more ambiguous. Among the standalone baselines,

Table 3: **MIT-Hard moiré-oriented evaluation.** SegDem suppresses residual moiré and false-color artifacts on high-frequency content, with the strongest advantage on RGGB and a smaller but consistent margin on Quad Bayer.

Method	RGGB					Quad Bayer				
	Linear RGB		sRGB			Linear RGB		sRGB		
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	PSNR↑	SSIM↑	LPIPS↓
Jd3Net	24.2812	0.7792	25.3435	0.7754	0.2443	22.4223	0.7375	24.0604	0.7391	0.2767
FFNet	24.5212	0.7218	23.9609	0.6934	0.3205	21.7905	0.5246	21.2632	0.4836	0.5158
ms-demosaic	27.6128	0.8304	27.5236	0.8081	0.2074	25.7716	0.8040	26.1715	0.7823	0.2427
BMTNet	27.3605	0.8221	27.9221	0.8056	0.2073	25.7552	0.8061	26.1562	0.7811	0.2432
SegDem(E)	27.8943	0.8586	27.6161	**0.8257**	**0.2040**	25.9452	0.8084	26.2920	0.7873	0.2384
SegDem(R)	**29.4532**	**0.8771**	**28.9201**	**0.8515**	**0.1667**	**26.8462**	**0.8213**	**26.6051**	**0.8007**	**0.2346**
SegDem(M)	**28.1354**	**0.8673**	**28.0143**	0.8237	0.2043	**25.9874**	**0.8103**	**26.3102**	**0.7892**	**0.2356**

ms-demosaic and BMTNet are the most competitive, whereas SegDem(R) obtains the best overall reconstruction on both patterns. Its advantage is largest on RGGB and smaller but consistent on Quad Bayer, indicating that the structural prior remains beneficial under stronger sampling ambiguity. The third row of Figure 6 likewise shows better preservation of repetitive structures with fewer residual moiré and false colors.

4.5 Ablation Study

We study structural transfer across backbone families and separately analyze reconstruction-side conditioning and representation alignment on EDMUNet.

Table 4: **Effect of structural pretraining across backbones.** Structural pretraining improves all three evaluated backbones, with a larger relative gain for EDMUNet in linear RGB and LPIPS, a stronger sRGB gain for Restormer, and smaller but consistent gains for MambaIR. Δ denotes Yes–No.

Backbone	Seg. pretrain	Single Bayer					Quad Bayer				
		Linear RGB		sRGB			Linear RGB		sRGB		
		PSNR↑	SSIM↑	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	PSNR↑	SSIM↑	LPIPS↓
EDMUNet	Yes	44.5753	0.9881	34.4157	0.9176	0.0832	43.1215	0.9858	33.8543	0.9224	0.0954
EDMUNet	No	43.6831	0.9848	33.5745	0.9054	0.1174	41.6736	0.9799	32.4886	0.9031	0.1425
EDMUNet	Δ	**+0.8922**	**+0.0033**	**+0.8412**	**+0.0122**	**-0.0342**	**+1.4479**	**+0.0059**	**+1.3657**	**+0.0193**	**-0.0471**
Restormer	Yes	45.5654	0.9908	35.1883	0.9312	0.0891	43.9275	0.9884	34.0871	0.9249	0.1061
Restormer	No	45.3144	0.9891	34.1242	0.9081	0.1007	43.7319	0.9879	33.1773	0.9038	0.1177
Restormer	Δ	**+0.2510**	**+0.0017**	**+1.0641**	**+0.0231**	**-0.0116**	**+0.1956**	**+0.0005**	**+0.9098**	**+0.0211**	**-0.0116**
MambaIR	Yes	45.5766	0.9907	35.1304	0.9327	0.0951	43.9385	0.9883	34.0355	0.9261	0.1129
MambaIR	No	45.4913	0.9879	34.5605	0.9301	0.0962	43.8559	0.9883	33.5241	0.9230	0.1144
MambaIR	Δ	**+0.0853**	**+0.0028**	**+0.5699**	**+0.0026**	**-0.0011**	**+0.0826**	**+0.0000**	**+0.5114**	**+0.0031**	**-0.0015**

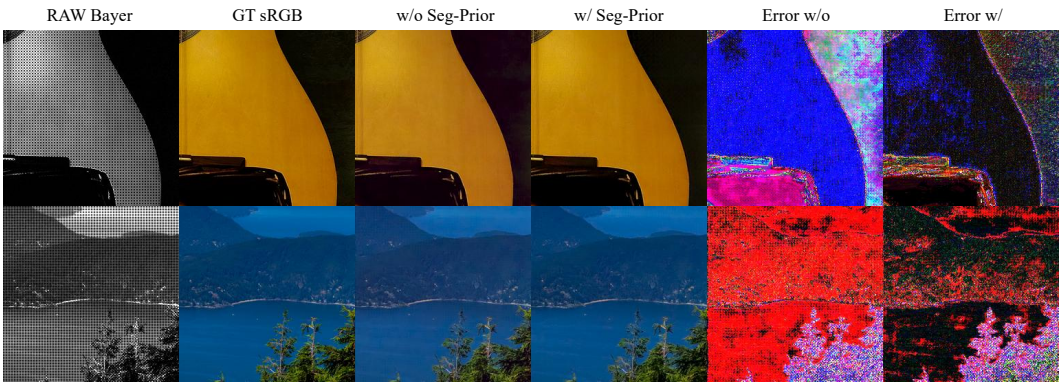


Figure 7: **Visual ablation of structural pretraining.** The error maps highlight regions where decoder transfer reduces structured color artifacts, especially around boundaries and repetitive textures.

Structural pretraining across backbones. Table 4 isolates the effect of structurally pretrained initialization, and Figure 7 provides the corresponding qualitative comparison. The gain is clearer

Table 5: **Ablation of reconstruction and sensor conditioning on EDMUNet.** The proxy provides a strong image-space starting point, whereas sparse RAW correction is especially important for preserving phase-sensitive evidence under Quad-Bayer sampling.

Variant	Single Bayer					Quad Bayer				
	Linear RGB		sRGB			Linear RGB		sRGB		
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	PSNR↑	SSIM↑	LPIPS↓
SegDem	**44.5753**	**0.9881**	**34.4157**	**0.9176**	**0.0832**	**43.1215**	**0.9858**	**33.8543**	**0.9224**	0.0954
w/o demosaicing proxy	40.5308	0.9865	33.0299	0.9088	0.0966	39.3487	0.9732	32.7300	0.9139	**0.0903**
w/ bilinear proxy	43.5412	**0.9873**	**34.0301**	**0.9130**	**0.0943**	**42.3497**	**0.9813**	**32.9612**	**0.9203**	**0.0895**
w/o RAW correction	**43.6812**	**0.9881**	33.8354	0.9035	0.1132	38.9717	0.9742	29.4347	0.8844	0.1829

Table 6: **Ablation of DINOv2 representation alignment on EDMUNet.** Alignment in both stages gives the most balanced result, while masked token alignment improves most PSNR and rendered-domain perceptual metrics.

Variant	Single Bayer					Quad Bayer				
	Linear RGB		sRGB			Linear RGB		sRGB		
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	PSNR↑	SSIM↑	LPIPS↓
SegDem	**44.5753**	**0.9881**	**34.4157**	**0.9176**	**0.0832**	**43.1215**	**0.9858**	**33.8543**	**0.9224**	**0.0954**
w/o DINOv2 in both stages	43.1578	0.9871	32.6838	0.8826	0.1210	42.7350	0.9829	32.4403	0.9029	0.1384
w/o DINOv2 in demosaicing	42.2790	0.9851	33.0503	**0.9124**	0.1002	41.6277	0.9819	32.9594	0.9118	0.1093
w/o DINOv2 in segmentation	43.6770	0.9849	33.8177	0.9016	0.1004	41.7929	0.9809	33.0630	0.9043	0.1165
w/o masked alignment	**44.4104**	**0.9895**	**34.3139**	0.9044	**0.0939**	**43.0261**	**0.9844**	**33.8345**	**0.9120**	**0.1049**

for EDMUNet than for Restormer in linear RGB and LPIPS: on Quad Bayer, EDMUNet improves by +1.4479 dB in linear PSNR and -0.0471 in LPIPS, whereas Restormer improves by +0.1956 dB and -0.0116, respectively. This suggests that the convolutional backbone benefits more from transferred local structure when recovering phase-sensitive details. Restormer's gains concentrate in rendered sRGB, where boundary color errors are more visible, while MambaIR improves more modestly. Overall, the prior transfers consistently across all three backbone families.

Reconstruction and sensor conditioning. Table 5 separates the components that determine how the model receives image-space and sensor-space evidence. Removing the demosaicing proxy produces a substantial drop on both CFA layouts, while bilinear interpolation recovers part of the lost performance but remains below the Jd3Net-based full model. Removing sparse RAW correction has a moderate effect on Single Bayer but reduces Quad-Bayer linear and sRGB PSNR by 4.1498 and 4.4196 dB, respectively. Thus, a strong proxy is not sufficient: the compact condition must also preserve phase information under Quad-Bayer sampling.

Representation alignment. Table 6 focuses on the training-time structural anchor. Removing DINOv2 alignment from both stages degrades reconstruction in both linear RGB and sRGB. Removing it from only segmentation or demosaicing also hurts performance, showing that the common reference is useful on both sides of the transfer rather than as a single-stage auxiliary loss. Aligning every token retains competitive linear SSIM but is worse on most PSNR and sRGB perceptual metrics, indicating that random masking provides a less restrictive regularizer.

5 Conclusion

This work presents SegDem, exploring the connection between visual understanding and image reconstruction. Our key premise is that both tasks describe the same underlying physical scene and should therefore preserve consistent structural information. We instantiate visual understanding with instance segmentation, using its region- and boundary-aware supervision to pretrain decoder representations that are subsequently transferred to RAW-conditioned demosaicing. A shared frozen DINOv2 representation further anchors structural consistency across the two stages, while explicit RAW conditioning preserves measurement fidelity. Experiments on Single- and Quad-Bayer demosaicing across three different backbones demonstrate the benefits of the proposed transfer framework. These results suggest that high-level visual understanding can provide useful structural representations for low-level reconstruction, offering a broader perspective on connecting understanding and reconstruction within a shared visual representation space.

AI USE STATEMENT

In this work, we used generative AI tools for drafting parts of the manuscript, editing the paper to improve readability and language, polishing the text, translating content, creating or modifying scientific figures or images, and formatting references. It's worth mentioning that we used GPT-Image 2 to show the visual comparison effects for the demosaicing task. We have not used generative AI tools for generating synthetic data sets, developing theoretical models or conceptual frameworks, formulating mathematical claims, proving mathematical claims, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, implementing methods, or interpreting results. We have reviewed all AI-assisted work. Specifically, all AI-generated text was checked for factual accuracy and potential plagiarism through a manual literature survey; all AI-assisted figures were verified and corrected by the authors to ensure they accurately represent the underlying data and methodology; and all AI-generated code was verified and tested for correctness by the authors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael G. Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 15619–15629. IEEE, 2023.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Bryce E. Bayer. Color imaging array, 1976. US Patent 3,971,065.

Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

Yan Chen, Yi Wen, Wei Li, Junchao Liu, Yong Guo, Jie Hu, and Xinghao Chen. RDDM: practicing RAW domain diffusion model for real-world image restoration. *CoRR*, abs/2508.19154, 2025.

Michaël Gharbi, Gaurav Chaurasia, Sylvain Paris, and Frédo Durand. Deep joint demosaicking and denoising. *ACM Trans. Graph.*, 35(6):191:1–191:12, November 2016. ISSN 0730-0301.

Ionut Grigore and Calin-Adrian Popa. Jepadepth: Masked predictive representation learning for self-supervised monocular depth estimation. *CoRR*, abs/2607.26600, 2026. doi: 10.48550/ARXIV.2607.26600. URL https://doi.org/10.48550/arXiv.2607.26600.

Hang Guo, Jinmin Li, Tao Dai, Zhihao Ouyang, Xudong Ren, and Shu-Tao Xia. MambaIR: A simple baseline for image restoration with state-space model, 2024.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *IEEE International Conference on Computer Vision*, 2017.

Om Khangaonkar and Hamed Pirsiavash. gen2seg: Generative models enable generalizable instance segmentation. *CoRR*, abs/2505.15263, 2025.

Filippos Kokkinos and Stamatios Lefkimmiatis. Deep image demosaicking using a cascade of convolutional residual denoising networks, 2018.

Saurabh Kumar and Nutan Sairam Yenneti. An extensible and lightweight unified architecture for demosaicing pixel-bin image sensors, 2026.

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

Haechang Lee, Dongwon Park, Wongi Jeong, Kijeong Kim, Hyunwoo Je, Dongil Ryu, and Se Young Chun. Efficient unified demosaicing for Bayer and non-Bayer patterned image sensors, 2023.

Yawei Li, Kai Zhang, Jingyun Liang, Jiezhang Cao, Ce Liu, Rui Gong, Yulun Zhang, Hao Tang, Yun Liu, Denis Demandolx, Rakesh Ranjan, Radu Timofte, and Luc Van Gool. LSDIR: A large scale dataset for image restoration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023 - Workshops, Vancouver, BC, Canada, June 17-24, 2023*, pp. 1775–1787. IEEE, 2023.

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 (ECCV)*, pp. 740–755. Springer, 2014.

Xinqi Lin, Jingwen He, Ziyan Chen, Zhaoyang Lyu, Bo Dai, Fanghua Yu, Wanli Ouyang, Yu Qiao, and Chao Dong. DiffBIR: Towards blind image restoration with generative diffusion prior, 2023.

Ziwei Luo, Fredrik K. Gustafsson, Zheng Zhao, Jens Sjölund, and Thomas B. Schön. Controlling vision-language models for multi-task image restoration, 2023.

Henrique S. Malvar, Li-wei He, and Ross Cutler. High-quality linear interpolation for demosaicing of Bayer-patterned color images. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2004.

Daniele Menon and Giancarlo Calvagno. Color image demosaicking: An overview. *Signal Process. Image Commun.*, 26(8-9):518–533, 2011.

Shentong Mo and Peter Tong. Connecting joint-embedding predictive architecture with contrastive self-supervised learning. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

Nilaksh, Saurav Jha, Artem Zholus, and Sarath Chandar. Reconstruction or semantics? what makes a latent space useful for robotic world models. *CoRR*, abs/2605.06388, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision, 2023.

Zhenghao Pan, Haijin Zeng, Jiezhang Cao, Yongyong Chen, Kai Zhang, and Yong Xu. Mambasci: Efficient mamba-unet for quad-bayer patterned video snapshot compressive imaging. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 10674–10685. IEEE, 2022.

Jaskirat Singh, Xingjian Leng, Zongze Wu, Liang Zheng, Richard Zhang, Eli Shechtman, and Saining Xie. What matters for representation alignment: Global information or spatial structure? *CoRR*, abs/2512.10794, 2025.

Weixiong Sun, Xiang Yin, and Chao Dong. Can nano banana 2 replace traditional image restoration models? an evaluation of its performance on image restoration tasks. *CoRR*, abs/2604.03061, 2026.

SaiKiran Kumar Tedla, Junyong Lee, Beixuan Yang, Mahmoud Afifi, and Michael S. Brown. Multispectral demosaicing via dual cameras. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pp. 5405–5414. IEEE, 2025a.

SaiKiran Kumar Tedla, Abhijith Punnappurath, Luxi Zhao, and Michael S. Brown. Examining joint demosaicing and denoising for single-, quad-, and nona-bayer patterns. In *IEEE International Conference on Computational Photography, ICCP 2025, Toronto, ON, Canada, July 21-23, 2025*, pp. 1–10. IEEE, 2025b.

Jianyi Wang, Zongsheng Yue, Shangchen Zhou, Kelvin C. K. Chan, and Chen Change Loy. Exploiting diffusion prior for real-world image super-resolution. In *IEEE/CVF International Conference on Computer Vision*, 2023.

Xiangming Wang, Benteng Sun, Yungeng Liu, Haijin Zeng, Yongyong Chen, Jingyong Su, and Jie Liu. HSI-VAR: rethinking hyperspectral restoration through spatial-spectral visual autoregression. *CoRR*, abs/2602.00749, 2026.

Rongyuan Wu, Lingchen Sun, Zhiyuan Ma, and Lei Zhang. One-step effective diffusion network for real-world image super-resolution, 2024a.

Rongyuan Wu, Lingchen Sun, Zhiyuan Ma, and Lei Zhang. SeeSR: Towards semantics-aware real-world image super-resolution. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024b.

Cuixin Yang, Rongkang Dong, and Kin-Man Lam. Vision-language model guided image restoration, 2025.

Qingyu Yang, Guang Yang, Jun Jiang, Chongyi Li, Ruicheng Feng, Shangchen Zhou, Wenxiu Sun, Qingpeng Zhu, Chen Change Loy, Jinwei Gu, Zhen Wang, Daoyu Li, Yuzhe Zhang, Lintao Peng, Xuyang Chang, Yinuo Zhang, Yaqi Wu, Xun Wu, Zhihao Fan, Chengjie Xia, Feng Zhang, Haijin Zeng, Kai Feng, Yongqiang Zhao, Hiêp Quang Luong, Jan Aelterman, Anh Minh Truong, Wilfried Philips, Xiaohong Liu, Jun Jia, Hanchi Sun, Guangtao Zhai, Longan Xiao, Qihang Xu, Ting Jiang, Qi Wu, Chengzhi Jiang, Mingyan Han, Xinpeng Li, Wenjie Lin, Youwei Li, Haoqiang Fan, Shuaicheng Liu, Rongyuan Wu, Lingchen Sun, and Qiaosi Yi. MIPI 2022 challenge on quad-bayer re-mosaic: Dataset and report. In Leonid Karlinsky, Tomer Michaeli, and Ko Nishino (eds.), *Computer Vision - ECCV 2022 Workshops - Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part V*, volume 13805 of *Lecture Notes in Computer Science*, pp. 21–35. Springer, 2022. doi: 10.1007/978-3-031-25072-9_2. URL https://doi.org/10.1007/978-3-031-25072-9_2.

Tao Yang, Rongyuan Wu, Peiran Ren, Xuansong Xie, and Lei Zhang. Pixel-aware stable diffusion for realistic image super-resolution and personalized stylization, 2023.

Yufeng Yang, Xianfang Zeng, Zhangqi Jiang, Fukun Yin, Jianzhuang Liu, Wei Cheng, Jinghong Lan, Shiyu Liu, Yuqi Peng, Gang Yu, and Shifeng Chen. Realrestorer: Towards generalizable real-world image restoration with large-scale image editing models. *CoRR*, abs/2603.25502, 2026.

Xiang Yin, Jinfan Hu, Zhiyuan You, Kainan Yan, Yu Tang, Chao Dong, and Jinjin Gu. How far have we gone in generative image restoration? a study on its capability, limitations and evaluation practices, 2026.

Weiyi You, Mingyang Zhang, Leheng Zhang, Xingyu Zhou, Kexuan Shi, and Shuhang Gu. Consistency trajectory matching for one-step generative super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12747–12756, October 2025.

F. Yu, J. Gu, Z. Li, J. Hu, X. Kong, X. Wang, J. He, Y. Qiao, and C. Dong. SUPIR: Scaling-up image restoration via generative prior. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Haijin Zeng, Kai Feng, Jiezhang Cao, Shaoguang Huang, Yongqiang Zhao, Hiep Quang Luong, Jan Aelterman, and Wilfried Philips. Inheriting bayer's legacy: Joint remosaicing and denoising for quad bayer image sensor. *Int. J. Comput. Vis.*, 132(11):4992–5013, 2024a.

Haijin Zeng, Hiep Luong, and Wilfried Philips. Wavelength-embedding-guided filter-array transformer for spectral demosaicing. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XIV*, volume 15072 of *Lecture Notes in Computer Science*, pp. 196–213. Springer, 2024b.

Haijin Zeng, Benteng Sun, Yongyong Chen, Jingyong Su, and Yong Xu. Spectral compressive imaging via unmixing-driven subspace diffusion refinement. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025a.

Haijin Zeng, Xiangming Wang, Yongyong Chen, Jingyong Su, and Jie Liu. Vision-language gradient descent-driven all-in-one deep unfolding networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 7524–7533. Computer Vision Foundation / IEEE, 2025b.

Donghui Zhang, Feiyu Li, Jun Yang, and Le Yang. Frequency-domain feature learning network for joint image demosaicing and denoising. *Mathematics*, 14(7), 2026. ISSN 2227-7390.

Tao Zhang, Ying Fu, and Cheng Li. Spatial adaptive network for real image demosaicing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 3326–3334, 2022.

Zhilu Zhang, Haolin Wang, Ming Liu, Ruohao Wang, Jiawei Zhang, and Wangmeng Zuo. Learning raw-to-srgb mappings with inaccurately aligned supervision. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pp. 4328–4338. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00431. URL https://doi.org/10.1109/ICCV48922.2021.00431.

Shiyang Zhou, Haijin Zeng, Yunfan Lu, Yongyong Chen, Jie Liu, and Jingyong Su. Lightweight quad Bayer HybridEVS demosaicing via state space augmented cross-attention, 2025a.

Shiyang Zhou, Haijin Zeng, Yunfan Lu, Tong Shao, Ke Tang, Yongyong Chen, Jie Liu, and Jingyong Su. Binarized mamba-transformer for lightweight quad bayer hybridevs demosaicing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 8817–8827. Computer Vision Foundation / IEEE, 2025b.

A Dataset Construction, Training, and Evaluation Details

We use the same paired data format for all demosaicing experiments. Each sample contains a single-channel RAW observation in `raw_lq`, a camera-linear RGB target in `lin_hq`, and a metadata record that stores the CFA pattern, color correction matrix, channel gains, gamma value, black and white levels, and noise parameters. The metadata is also used by the fixed PTP operator $\Gamma(\cdot)$, so the sRGB losses and sRGB evaluation metrics are computed from the same rendering transform.

Synthetic demosaicing pairs. The synthetic demosaicing benchmark is constructed from LSDIR (Li et al., 2023). We use an image-level 7:1:2 train/validation/test split, so images used for validation or testing are excluded from demosaicing training. We first resize and crop images to 512×512, convert display sRGB images to camera-linear RGB with the inverse PTP transform, and then sample the linear RGB image according to the target CFA layout. For Single-Bayer data, we use the standard RGGB arrangement. For Quad-Bayer data, we use a 4×4 repeating layout with 2×2 same-channel blocks. The inverse PTP uses a fixed mobile-camera color profile with gamma 2.2, a fixed 3×3 color correction matrix, and fixed channel gains. After mosaicing, we add light signal-independent RAW Gaussian perturbation with $\sigma \sim \mathcal{U}(0, 0.01)$. This perturbation is kept small and is shared across all compared methods; the task remains demosaicing from CFA measurements to clean linear RGB. During training, 512×512 crops are sampled with even pixel offsets to preserve CFA phase. During validation and testing, center crops are used.

Real RAW evaluation data. We evaluate real-data generalization on two unpaired RAW benchmarks. The Single-Bayer benchmark uses the Huawei split of the Zurich RAW-to-RGB (ZRR) dataset, containing 1204 native RGGB frames at 448×448 resolution. The raw observations are stored as uint16 PNG sensor readings. We use a black level of 64 and white level of 1023, subtract the black level, and normalize each RAW observation as $x_{\text{norm}} = \text{clip}((x - \text{black})/(\text{white} - \text{black}), 0, 1)$. The dataset-provided target images are produced by the manufacturer ISP and are used only to verify spatial alignment, not as clean pixel-level references. We therefore evaluate the rendered predictions with no-reference IQA rather than PSNR or SSIM.

The Quad-Bayer benchmark uses the MIPI 2022 Quad-Bayer Re-mosaicing training split at the 0 dB noise setting. It contains 70 native Quad-Bayer frames at 1800×1200 resolution with 2×2 same-channel blocks following the block-level RGGB order. The black and white levels are 64 and 1023, respectively. Each frame is accompanied by image metadata including red/blue gains and a 3×3 color correction matrix. The provided Bayer-domain reference is aligned with the Quad-Bayer input, but it is a single-channel Bayer image rather than a clean linear RGB target; it is therefore used only for sanity checking and visualization, not as a demosaicing ground truth.

Because the MIPI split is dark and has large frame-to-frame exposure variation, we apply a frame-wise exposure normalization before model inference. Specifically, the green-channel median of each frame is scaled toward the training-domain median of 0.135, with the exposure scale clamped to $[1, 8]$. The full-resolution MIPI frames are tiled into 512×512 patches with stride 256 and stitched back by averaging overlapping regions. All patches from the same original frame share the same exposure compensation factor to avoid brightness seams after stitching. The final IQA scores are computed on stitched full-resolution sRGB outputs.

Real-data rendering and IQA evaluation. For both real RAW datasets, all methods share the same deterministic rendering pipeline $\Gamma(\cdot)$. The predicted linear RGB is first white-balanced with the metadata gains, transformed by the corresponding color correction matrix, multiplied by the shared exposure compensation, mapped through a gamma-2.2 tone curve, and finally adjusted with a fixed saturation factor of 1.20. Since no clean linear RGB reference is available, we report NIQE, MUSIQ, MANIQA, TOPIQ, and AFINE-NR on the rendered sRGB predictions. AFINE-NR requires image dimensions divisible by 32; therefore, MIPI full-resolution predictions are center-cropped to 1792×1184 for AFINE-NR only. Each metric is computed from a cloned input tensor to avoid side effects from metrics that internally modify the input range.

Segmentation pretraining data. The segmentation stage uses COCO instance masks under the same image-level 7:1:2 split. Each COCO image and its masks are resized and center-cropped to 512×512, converted to camera-linear RGB by the same inverse PTP, and mosaiced into both RGGB

and Quad Bayer RAW observations. The instance masks are kept as COCO-style annotations and converted to RGB-coded instance targets for the auxiliary instance-code head. They are used only in segmentation pretraining and are not used as demosaicing targets or during demosaicing inference.

SANet-derived evaluation data. For real-reference evaluation, we use the clean split of the SANet real image demosaicing dataset (Zhang et al., 2022). The SANet ARQ files provide pixel-shift camera-linear RGB references. We center-crop each reference image to the evaluation size and re-mosaic it with RGGB and Quad Bayer CFAs to obtain the corresponding RAW observations, producing noise-free paired samples for pure demosaicing evaluation. These SANet-derived pairs are reserved for evaluation only. The metadata records the ARQ source, crop position, CFA pattern, camera white balance derived from the ARQ file, identity color correction matrix, gamma 2.2, and zero synthetic noise.

Training protocol. SegDem is trained in two stages for each reconstruction backbone. In the first stage, the selected backbone, RAW correction module, instruction adapters, representation head, and RGB-coded instance head are trained on the COCO RAW instance data using the segmentation instruction. This stage is not intended to solve segmentation directly from incomplete RAW measurements: the image state is the clean linear RGB target, while the RAW condition remains packed from the mosaicked observation to preserve the sensor-conditioning interface used in Stage 2. The instance loss weight is $\lambda_{\text{inst}} = 1.0$ and the segmentation-stage DINO weight is $\lambda_{\text{repr}}^{\text{seg}} = 0.1$. The DINO target is extracted from the fixed-PTP rendering $\Gamma(x)$ of the clean linear RGB image. The DINOv2 encoder, the instruction encoder, and the frozen Jd3Net demosaicing proxy are not updated. The optimizer is AdamW with learning rate 5×10^{-5}, weight decay 10^{-4}, gradient clipping at 1.0, and bfloat16 mixed precision.

In the second stage, the corresponding demosaicing model is initialized from its structurally pretrained checkpoint and fine-tuned on the LSDIR RAW demosaicing pairs for 10 epochs using the demosaicing instruction. The demosaicing-only baseline uses the same demosaicing training data and schedule without segmentation initialization. Training uses AdamW with learning rate 2×10^{-5}, zero weight decay, gradient clipping at 1.0, bfloat16 mixed precision, and distributed data parallel training when multiple GPUs are used. The demosaicing objective uses Charbonnier linear RGB loss, ℓ_1 sRGB loss after PTP, LPIPS loss in sRGB space, edge loss, and decoder-token DINO alignment with weights 1.0, 1.0, 0.2, 0.1, and 0.1, respectively. The noise-conditioning scalar is fixed to $\sigma = 0.01$ for the EDMUNet instantiation during demosaicing.

Evaluation protocol. Synthetic evaluation uses the held-out test split from the 7:1:2 image-level partition. Validation images are used only for checkpoint selection, and test images are not used in segmentation pretraining, demosaicing training, or model selection. For real captured RAW evaluation, we render each method output to sRGB and report no-reference IQA metrics because clean linear RGB references are unavailable. The SANet-derived paired evaluation is reported in Appendix B. For paired synthetic and SANet-derived evaluation, we report PSNR and SSIM in linear RGB by directly comparing the predicted linear RGB image with `lin_hq`. We also report PSNR, SSIM, and LPIPS in rendered sRGB by applying the metadata-defined PTP operator to both prediction and target. This keeps the linear and sRGB metrics tied to the same paired linear RGB target, while avoiding any learned or image-dependent ISP during evaluation.

B Additional SANet-Derived Evaluation

We include SANet-derived paired evaluation as an additional external reference experiment. The ARQ pixel-shift linear RGB references from the SANet real image demosaicing dataset (Zhang et al., 2022) are cropped to the evaluation size and re-mosaicked with RGGB and Quad Bayer CFA layouts, yielding an external noise-free pure-demosaicing test set that is not used for training or model selection.

Table 7 shows that the gains also hold in this SANet-derived pure-demosaicing setting. This result is reported in the appendix because the main real-data evaluation focuses on camera-captured RAW images without clean references.

Table 7: **Additional SANet-derived evaluation.** The transferred decoder prior generalizes to external pairs, with the largest gains appearing after sRGB rendering and Quad Bayer remaining the harder setting.

Method	RGGB					Quad Bayer				
	Linear RGB		sRGB			Linear RGB		sRGB		
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	PSNR↑	SSIM↑	LPIPS↓
SANet	52.7426	0.9794	42.3979	0.9342	0.2601	52.1435	0.9786	42.0101	0.9211	0.2832
Jd3Net	53.8407	0.9891	42.0679	0.9434	0.2703	53.7631	0.9876	42.1325	0.9415	0.2631
ms-demosaic	55.1885	0.9915	42.8805	0.9473	0.2096	54.7635	0.9902	42.6425	0.9385	0.2863
BMTNet	55.3262	0.9916	43.2509	0.9401	0.2094	52.1638	0.9821	40.3251	0.9311	0.3011
FFNet	55.1670	0.9917	42.9577	0.9478	0.2516	46.3257	0.9831	36.2578	0.9308	0.3216
SegDem(E)	55.4347	0.9924	43.7238	0.9491	0.2017	55.3304	**0.9926**	43.7693	0.9477	0.2586
SegDem(R)	**56.1340**	**0.9943**	**45.6526**	**0.9710**	**0.0902**	**55.5101**	**0.9926**	**44.4649**	**0.9519**	**0.1828**
SegDem(M)	**56.1431**	**0.9942**	**45.6752**	**0.9702**	**0.0911**	**55.4612**	**0.9931**	**44.5102**	**0.9712**	**0.1817**

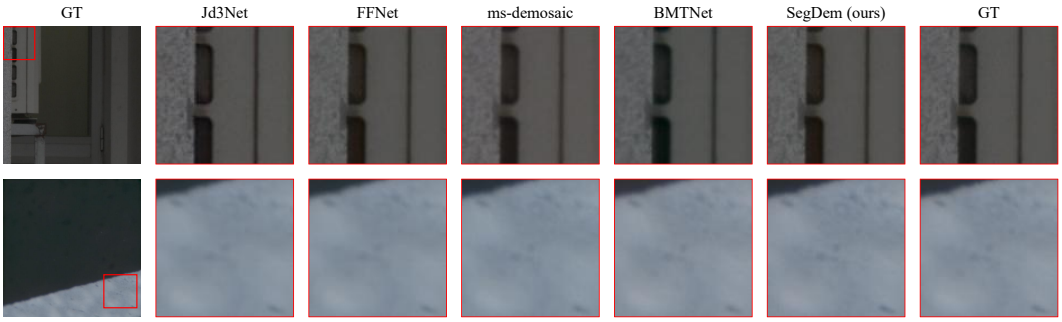


Figure 8: **SANet-derived captured RAW visual comparison.** The examples compare rendered sRGB outputs on SANet-captured RGGB and Quad-Bayer RAW images.

C Framework Instantiation Details

SegDem is defined by a common RAW-conditioned reconstruction interface rather than by a single backbone. Each instantiation contains four parts: an image-state branch, a RAW condition branch, task-conditioned auxiliary adapters, and a representation head. Unless otherwise specified, all main experiments use the frozen Jd3Net output as the coarse linear-RGB reconstruction state. SegDem refines this state using the explicit packed-RAW condition and the structurally pretrained shared decoder. The RAW condition branch receives the packed CFA condition described in Appendix D. The adapters produce cached task-specific auxiliary features for instance-code supervision and DINOv2 alignment.

Under this interface, the restoration backbones differ only in how they process features and inject the RAW condition. We instantiate SegDem with convolutional encoder–decoder, Transformer-based, and state-space restoration backbones to test whether the transferred decoder representation is tied to a particular architecture.

Convolutional encoder-decoder instantiation. The convolutional instantiation uses EDMUNet as the reconstruction backbone. The image branch receives the reconstruction state, while the RAW branch receives the corrected packed CFA condition. The condition is injected into the U-Net feature pathway through convolutional condition projections. Decoder features are exposed through selected decoder blocks for instruction modulation and representation alignment. The network outputs camera-linear RGB in one forward pass. Although this backbone follows a noise-conditioned restoration parameterization, we use a fixed scalar during demosaicing and do not run an iterative sampling process.

Transformer restoration instantiation. The Transformer instantiation uses Restormer as the reconstruction backbone. We keep the high-resolution encoder-decoder design and modify only the input, condition, decoder adaptation, and auxiliary representation heads. The image-state branch uses an overlapping patch embedding, followed by four Transformer levels with channel dimen-

sions 48, 96, 192, and 384. The numbers of Transformer blocks are $[4, 6, 6, 8]$, and the corresponding attention heads are $[1, 2, 4, 8]$. The decoder mirrors the encoder with skip connections, channel reduction after concatenation, three upsampling stages, and four refinement Transformer blocks.

The Transformer condition path receives the four-channel packed RAW condition. A bounded correction module first predicts a residual with three 3×3 convolutional layers and a $\tanh$ bound of 0.25. The corrected condition is resized to the image resolution, embedded by two 3×3 convolutions, and added to the patch-embedded reconstruction state with a learnable scalar. This early fusion gives the Transformer blocks access to the sensor constraint while keeping the RAW condition separate from the image-space reconstruction state.

Instruction adapters are inserted after the third, second, and first decoder stages and after the refinement stage. Each adapter follows the FiLM-bottleneck structure in Appendix E. The representation head consumes three adapted decoder features, projects them with lateral 1×1 convolutions, fuses them in an FPN-style top-down path, and optionally predicts a 16×16 grid of 768-dimensional DINO-aligned tokens. During the segmentation stage, the same fused representation also produces an auxiliary RGB-coded instance map. During demosaicing inference, the auxiliary head is not used as an output.

The final Transformer prediction is a residual over the reconstruction state. The output convolution predicts a three-channel correction, which is added to the Jd3Net proxy input and clamped to the valid linear RGB range. This residual formulation makes the backbone focus on demosaicing errors left by the coarse reconstruction state, while the packed RAW condition keeps the reconstruction tied to the measured CFA samples.

State-space restoration instantiation. The state-space instantiation uses MambaIR (Guo et al., 2024) as the restoration backbone. We keep its shallow convolutional embedding, residual state-space groups, patch embedding/unembedding operators, and final restoration head, and modify only the same interface components used by the other SegDem backbones. The image-state branch receives the coarse linear-RGB reconstruction state. The packed RAW condition is corrected by the bounded RAW correction module, embedded by two 3×3 convolutions, and added to the shallow MambaIR feature before the residual state-space groups.

Each residual state-space group exposes a spatial feature map by unembedding the group tokens back to the image grid. We attach the same FiLM-bottleneck instruction adapter to these group features and then re-embed the adapted feature before the next group. The representation head consumes the last three adapted group features, fuses them with the same FPN-style lateral projections, and predicts the 16×16 DINO-aligned token grid. When the auxiliary instance-code head is enabled, it is attached to this fused representation in the same way as the other instantiations.

The final MambaIR prediction follows the residual restoration form of the original backbone: the network predicts a correction over the input reconstruction state and outputs camera-linear RGB. Thus the MambaIR variant changes the restoration operator from convolution/attention to selective state-space modeling, while keeping RAW conditioning, task-conditioned auxiliary features, decoder transfer, and inference-time outputs consistent with EDMUNet and Restormer.

D Detailed Components of Condition Branches

Figure 9 shows the detailed condition and representation-target construction used by SegDem. The two branches serve different roles. The sensor condition branch builds a measurement-aware RAW condition for linear RGB reconstruction, while the representation target branch builds training-only semantic targets used to regularize decoder features.

Sensor Condition Branch. Given a RAW Bayer observation, the sensor condition branch first applies pattern-aware sparse CFA packing. For Single-Bayer inputs, the RGGB samples are separated into four sparse planes corresponding to R, G_1, G_2, and B, giving $c_{\mathrm{pack}} \in \mathbb{R}^{4 \times H/2 \times W/2}$. For Quad-Bayer inputs, the 2×2 same-channel CFA blocks are converted into the same RGGB-like four-plane representation. This design gives different CFA layouts a shared condition format while preserving the location of observed sensor samples.

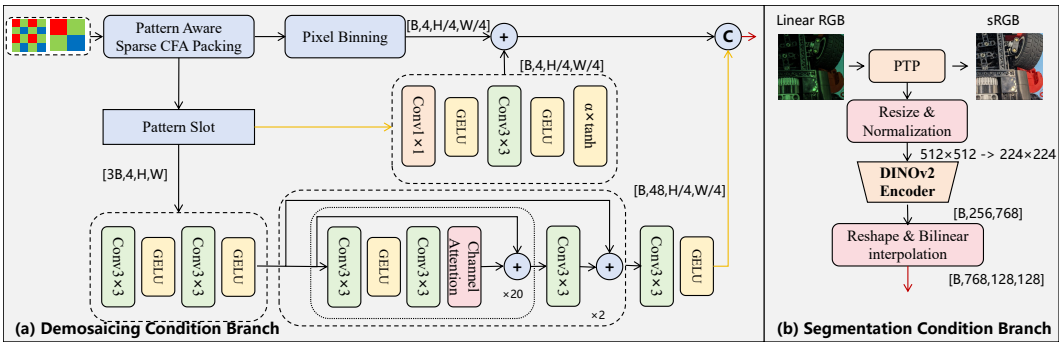


Figure 9: **Detailed structure of the sensor condition and representation target branches.** The sensor condition branch converts Bayer observations into a packed RAW condition with pattern-aware sparse CFA packing, pixel binning, and sparse RAW correction. The representation target branch renders linear RGB into sRGB with the fixed PTP operator and extracts dense DINOv2 tokens for training-only representation alignment.

The $[3B, 4, H, W]$ annotation in Figure 9 denotes an implementation-level buffer inherited from the Jd3Net condition interface. That interface reserves three RGGB-like pattern groups for compatibility with Nona-Bayer inputs and temporarily folds the group axis into the batch axis. The present work evaluates only Single- and Quad-Bayer data; after pattern-specific group selection and pixel binning, the effective per-sample condition used by SegDem is $c_{\text{bin}} \in \mathbb{R}^{4\times H/4\times W/4}$ with batch size B. Thus, the $3B$ label describes the general implementation buffer rather than three independent samples or a different condition dimensionality in the experiments reported here.

The packed sparse planes are then reduced by pixel binning to form a compact low-resolution RAW condition $c_{\text{bin}} \in \mathbb{R}^{4\times H/4\times W/4}$. This step is efficient, but it can also blur the difference between CFA phase, local sampling density, and high-frequency sensor evidence. The problem is particularly severe for Quad Bayer inputs. Each color is observed in 2×2 same-channel blocks, so binning can collapse several phase-dependent measurements into a similar low-resolution code. The demosaicing backbone then receives a condition that preserves the average color evidence but weakens where and how each sensor sample was observed.

The sparse RAW correction module is designed to recover this missing phase-aware information before the condition is consumed by the backbone. It receives a pattern-slot representation that explicitly encodes CFA phase information, processes it with convolutional blocks and channel attention, and predicts a bounded residual for the binned RAW condition. This residual does not replace the measured RAW values. Instead, it corrects packing and binning artifacts while keeping the condition tied to the observed CFA structure. The corrected RAW condition is concatenated with the projected correction features to produce the final demosaicing condition c_{raw}, which is injected into the one-step RAW-conditioned demosaicing backbone.

The ablation in Table 5 confirms the importance of this module. Removing sparse RAW correction has a modest effect on Single Bayer, but it severely degrades Quad Bayer reconstruction, reducing linear PSNR by 4.1498 dB and sRGB PSNR by 4.4196 dB. This gap supports our design choice. Quad Bayer demosaicing is more sensitive to phase loss introduced by compact condition construction, and an explicit phase-aware residual correction is needed to preserve the sensor layout information required for accurate color reconstruction.

Representation Target Branch. The representation target branch constructs semantic targets for decoder representation alignment. Since DINOv2 operates on rendered RGB images, the ground-truth linear RGB image is first mapped to sRGB by the fixed PTP operator $\Gamma(\cdot)$. This operator applies metadata-defined channel gains, a 3×3 color correction matrix, clamping, and gamma correction. The rendered image is then resized and normalized before being processed by the frozen DINOv2 encoder.

The DINOv2 patch tokens are reshaped into a dense spatial grid and resized to match the decoder-token resolution. These tokens serve as z_{dino} in Eq. 5. During training, decoder features are pro-

jected to z_θ and aligned with z_{dino}, providing a spatial-semantic anchor for decoder features without adding a segmentation output head during demosaicing inference.

E Instruction Adapter Details

Figure 10 illustrates the instruction adapter used to make the shared decoder representation task-aware. The adapter is intentionally lightweight. It does not concatenate language tokens with the RAW input and does not introduce an autoregressive text-to-image generation path. Instead, it converts a fixed task instruction into channel-wise modulation parameters for selected auxiliary decoder features.

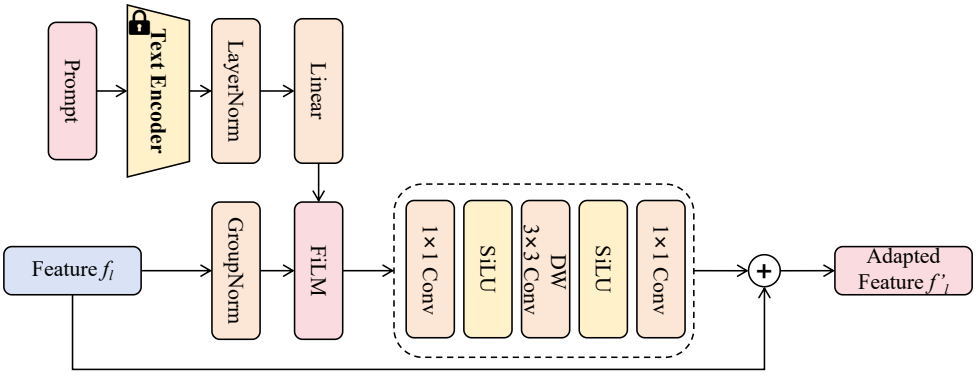


Figure 10: **Instruction adapter.** A frozen Qwen2.5-VL encoder maps a task prompt into a cached instruction embedding. The embedding is projected to FiLM scale and shift parameters, which produce task-conditioned auxiliary decoder features for instance-code supervision and token alignment.

We define two task prompts. The demosaicing prompt is "Recover clean linear RGB from noisy Single-Bayer or Quad-Bayer RAW observations. Preserve edges, color fidelity, and high-frequency details for demosaicing." The segmentation prompt is "Segment every visible COCO object instance from the RAW Bayer scene state. Convert the RAW-conditioned U-Net representation into a mask-centric region representation with instance-level boundaries." Each prompt is encoded once by a frozen Qwen2.5-VL encoder (Bai et al., 2025), and the resulting embedding is cached and reused during training and evaluation. Therefore, the vision-language model is not updated by the demosaicing objective and is not repeatedly executed during normal training.

For a decoder feature $f_\ell \in \mathbb{R}^{C_\ell \times H_\ell \times W_\ell}$ at level ℓ, the cached instruction embedding e_{task} is first normalized and linearly projected to FiLM parameters,

$$[\gamma_\ell, \beta_\ell] = W_\ell \text{LN}(e_{\text{task}}), \tag{7}$$

where $\gamma_\ell, \beta_\ell \in \mathbb{R}^{C_\ell}$. The feature modulation is applied after group normalization,

$$\tilde{f}_\ell = \text{GN}(f_\ell) \odot (1 + 0.1 \tanh(\gamma_\ell)) + \beta_\ell. \tag{8}$$

The bounded scale term prevents large instruction-induced amplification at the beginning of fine-tuning, while the shift term provides task-specific feature re-centering.

For the auxiliary representation branch, the modulated feature is processed by a residual bottleneck adapter,

$$f_\ell^{\text{out}} = f_\ell + \alpha_\ell A_\ell(\tilde{f}_\ell), \tag{9}$$

where A_ℓ is a 1×1 convolution, SiLU activation, depthwise 3×3 convolution, SiLU activation, and another 1×1 convolution. The hidden width is $C_\ell/4$ with a minimum of 16 channels, and the residual scale α_ℓ is initialized to 10^{-3}. We attach adapters to the decoder levels used by the feature pyramid head, namely 16×16, 32×32, 64×64, and 128×128 decoder features. These levels cover global layout, object boundaries, and local texture details.

The adapter serves different roles in the two stages. During structural pretraining, the segmentation prompt produces instance-code-oriented auxiliary features consumed by the RGB-coded instance

head. This encourages the shared decoder to organize its representation around object regions and boundaries. During demosaicing, the demosaicing prompt produces reconstruction-oriented auxiliary features for DINOv2 token alignment. The sensor-conditioned reconstruction path remains unchanged by the language encoder and continues to be supervised by paired linear RGB targets and constrained by the packed CFA condition. In the implementation, decoder forward hooks cache the adapted features for auxiliary token alignment and task heads. The instruction therefore selects the active auxiliary representation objective without allowing language conditioning to override the measured RAW evidence.

F Other Implementation Details

F.1 DINOv2 Tokens

For representation alignment, the target linear RGB image is first rendered by the fixed PTP operator and resized to 224×224. We then apply the standard ImageNet normalization used by DINOv2 and feed the image to a frozen DINOv2-base encoder (Oquab et al., 2023). The class token is discarded, and the remaining patch tokens form a 16×16 spatial grid with 768 channels. These tokens are used only as supervision targets. Gradients are not propagated into the DINOv2 encoder.

The reconstruction decoder produces a token grid with the same spatial size and channel dimension through the representation head. For the EDMUNet instantiation, both segmentation and demosaicing stages use masked DINO token alignment. We sample a new random token mask for each batch with mask ratio 0.4 over the 16×16 token grid. The mask is independently sampled per image; selected tokens are included in the Smooth L1 token loss after token-wise layer normalization, while unselected tokens are excluded from the loss rather than zeroed. This masked alignment is used only during training. At inference, the DINOv2 encoder, token targets, and random mask are completely removed.

F.2 Segmentation Head Details

The segmentation stage is used to learn decoder representations, not to define an extra output for demosaicing. Inspired by Gen2Seg (Khangaonkar & Pirsiavash, 2025), we formulate instance-level supervision as RGB-coded image-to-image prediction rather than conventional instance segmentation. The adapted decoder features are fused by the representation head, and an auxiliary head predicts a three-channel instance-code map $\hat{q} \in [0,1]^{3\times H\times W}$. COCO instance masks are converted to a deterministic RGB target q: background pixels are black; each object instance receives a deterministic color code; larger instances are drawn first and smaller instances overwrite overlaps so small objects remain visible.

The instance loss is computed on this RGB-coded map,

$$\mathcal{L}_{\text{inst}} = \mathcal{L}_{\text{rgb}} + \lambda_{\text{rgb-edge}}\mathcal{L}_{\text{edge}}(\hat{q}, q) + \lambda_{\text{emb}}\mathcal{L}_{\text{emb}} + \lambda_{\text{dice}}\mathcal{L}_{\text{fg-dice}} + \lambda_{\text{focal}}\mathcal{L}_{\text{fg-focal}} + \lambda_{\text{bd}}\mathcal{L}_{\text{fg-boundary}}. \tag{10}$$

$\mathcal{L}_{\text{rgb}}$ is a Charbonnier RGB regression loss with separate foreground and background terms, where the background term is weighted by 0.2. $\mathcal{L}_{\text{emb}}$ is a Gen2Seg-style discriminative embedding loss on the predicted three-channel code: pixels from the same instance are pulled toward their instance mean, different instance means are repelled, and background pixels are encouraged to remain near zero. The edge, foreground Dice, focal, and boundary terms stabilize object extent and boundary learning. We use $\lambda_{\text{rgb-edge}} = 0.25$, $\lambda_{\text{emb}} = 1.0$, $\lambda_{\text{dice}} = 1.0$, $\lambda_{\text{focal}} = 1.0$, and $\lambda_{\text{bd}} = 0.25$.

This head is therefore different from traditional Mask R-CNN-style instance segmentation (He et al., 2017): it has no class prediction, proposal matching, mask queries, or one binary mask per instance. Dice and foreground IoU are computed from the predicted RGB code map by thresholding the maximum RGB channel, $\hat{f} = \mathbf{1}[\max_c \hat{q}_c > 0.08]$, and comparing it with the union of COCO instance masks. These metrics are reported only to verify that the segmentation pretraining stage learns meaningful region structure. The segmentation head is discarded during demosaicing.

During segmentation pretraining, the reconstruction state is the paired clean linear RGB image, while the RAW condition remains packed from the mosaicked observation. This prevents the segmentation objective from being dominated by demosaicing errors and focuses the stage on learning

object-region and boundary representations. During demosaicing training and inference, the segmentation head is inactive and no mask prediction is produced.

F.3 Demosaicing Head Details

The demosaicing head always predicts camera-linear RGB. For EDMUNet, the output is produced by the original convolutional reconstruction head. For Restormer and MambaIR, the output convolution predicts a three-channel residual that is added to the reconstruction state. The final prediction is clamped to $[0, 1]$ before losses and metrics are computed. This residual design is used only to stabilize restoration around the proxy reconstruction; the packed RAW condition still provides the sensor constraint.

The reconstruction state is generated consistently for both training and evaluation. We use a frozen Jd3Net demosaicing proxy (Tedla et al., 2025b); the sparse RAW input is mapped to the proxy model's expected range. In our experiments this proxy input uses the $[-1, 1]$ range, while RAW values and linear RGB targets are stored in $[0, 1]$. For Single-Bayer and Quad-Bayer inputs, the appropriate proxy output branch is selected according to the CFA pattern and resized to the RAW resolution if needed.

The sRGB loss and sRGB metrics are computed by the same fixed PTP operator $\Gamma(\cdot)$. We do not train a learned ISP and do not use image-dependent tone mapping during evaluation. Therefore, improvements in sRGB metrics reflect demosaicing errors that remain after a fixed rendering transform, especially false colors and boundary artifacts.

G Supplementary Instance Segmentation Evaluation

Although the main paper focuses on RAW demosaicing, we also evaluate the structural pretraining stage on COCO instance masks to verify that the decoder learns meaningful region and boundary representations while retaining the RAW-conditioning interface used for subsequent demosaicing.

Table 8: **COCO instance segmentation evaluation.**

Strategy	Dice ↑	FG-IoU ↑
Seg. only	0.7643	0.6941
Demo. init + Seg.	0.7682	0.6969

Table 8 compares two training strategies for the segmentation stage. "Seg. only" trains the conditional backbone and the RGB-coded instance head only with the segmentation objective, while "Demo. init + Seg." initializes the same segmentation model from demosaicing pretraining and then fine-tunes it for instance-code supervision. The gain is moderate but consistent, indicating that RAW reconstruction pretraining learns boundary-sensitive and region-aware decoder representations that remain compatible with instance-level semantic grouping.

Figure 11 provides a qualitative view of the same comparison. Both models recover the dominant foreground regions, showing that the decoder learns instance-level region structure while remaining compatible with the mosaicked-observation condition used in Stage 2.

H Limitations

SegDem uses instance segmentation as a concrete instantiation of high-level visual understanding to study whether structurally informed representations can benefit low-level CFA reconstruction. We do not claim that instance segmentation is the uniquely optimal source of high-level supervision. Other understanding tasks, such as depth estimation, semantic segmentation, or visual correspondence, may provide complementary structural cues. Exploring a broader family of high-level representations and their transferability to demosaicing is an interesting direction for future work.

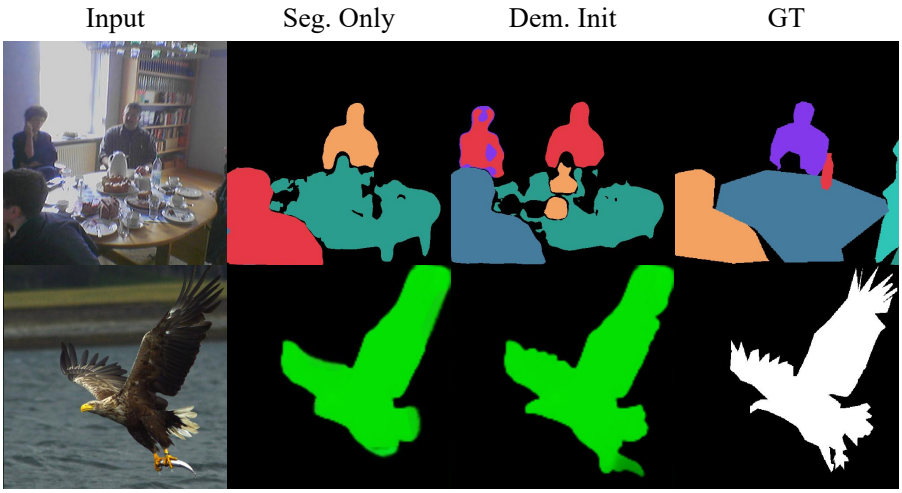


Figure 11: **Supplementary segmentation visual comparison.** From left to right, we show the rendered input, the segmentation-only prediction, the demosaicing-initialized prediction, and the ground-truth instance mask.

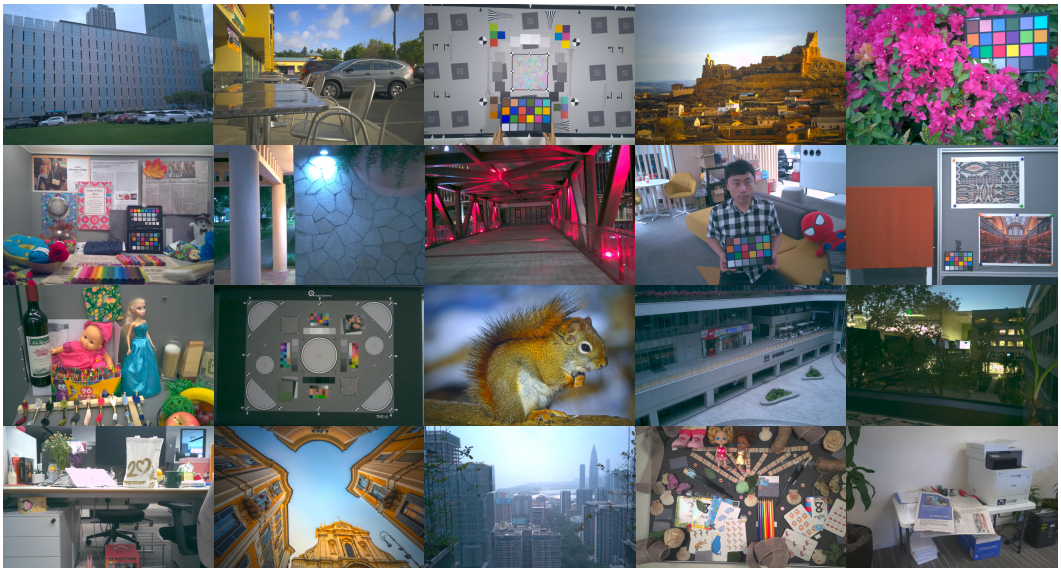


Figure 12: **More Visual Results in Real RAW Datasets.**