SUPER-RESOLVING UNSEEN HYPERSPECTRAL SENSORS AT ANY SCALE VIA SPATIAL OPERATORS

Anonymous authors
Paper under double-blind review

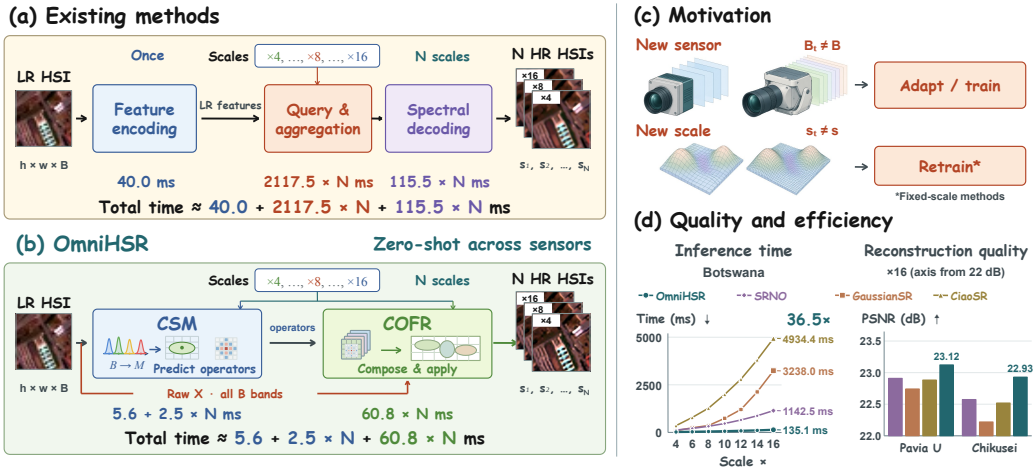


Figure 1: **Cross-sensor, arbitrary-scale reconstruction with OmniHSR.** (a) Existing methods, here CiaoSR, decode the spectral values of every band from features and repeat querying and decoding per requested scale. (b) OmniHSR predicts local spatial operators with CSM and applies them to all observed bands with COFR. (c) A new sensor requires adapting or retraining existing methods, and a new scale requires retraining fixed-scale methods. (d) At $\times 16$, OmniHSR reaches a higher zero-shot PSNR than the baselines on Pavia U and Chikusei and runs $36.5\times$ faster than CiaoSR on the 145-band Botswana. N is the number of requested scales; times are measured on Botswana.

ABSTRACT

Achieving cross-sensor generalization and arbitrary-scale reconstruction with a single model remains challenging in hyperspectral super-resolution (HSR). Although recent methods support arbitrary-scale reconstruction, applying them to new sensors or scales beyond the training range often requires additional data and computation to maintain reconstruction quality. To address these challenges, we propose OmniHSR, which predicts band-shared spatial operators rather than spectral values. Cross-Spectral Mapping (CSM) resamples inputs with any number of bands to fixed reference positions and predicts local operators with Gaussian supports. Continuous Operator-Field Reconstruction (COFR) composes these operators into a continuous field and applies them to all original bands for arbitrary-scale reconstruction. Experiments demonstrate that operator prediction outperforms direct spectral-value prediction on all seven datasets. Trained solely on ARAD with only 0.538M parameters, OmniHSR outperforms all directly transferred baselines on six unseen datasets without target-domain training data or adaptation. Across twelve upsampling factors from $\times 2$ to $\times 48$, it improves average PSNR on Pavia U and Chikusei by 0.55 dB over the strongest baseline. It also surpasses baselines trained from scratch or adapted on the target sensor and achieves up to $36\times$ faster inference. Our code will be publicly released soon.

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.

1 INTRODUCTION

Hyperspectral images provide rich spectral information but often lack sufficient spatial detail for practical applications, motivating hyperspectral image super-resolution (HSR) (Bian et al., 2024; Li et al., 2025; Ciotola et al., 2025; Guo et al., 2023). Most existing HSR methods are designed for fixed upsampling factors such as $\times 2$, $\times 4$, and $\times 8$ (Wang et al., 2022; Zhang et al., 2023; Chen et al., 2024b; 2025b), while recent arbitrary-scale methods handle different factors with a single model (Chen et al., 2021; Hu et al., 2025; Jiang et al., 2024; Chen et al., 2024a). Hyperspectral sensors also differ in their band configurations, from 31 to more than one hundred bands, and the public data of a remote-sensing sensor is often a single scene. Achieving cross-sensor generalization and arbitrary-scale reconstruction with a single model remains challenging.

Existing HSR networks output the spectral values of every band, with as many output channels as the training sensor has bands (Figure 1a). To serve a new sensor, they either run on sliding windows of bands, which multiplies the inference cost with the band count, or are trained from scratch or adapted on the target sensor (Scheibenreif et al., 2024; Thoreau et al., 2025), which needs target data (Figure 1c). Beyond their training range of factors, fixed-scale methods must be retrained and arbitrary-scale methods lose accuracy. In all these cases, the network itself still predicts the spectral values, for every band and at every factor.

To address these challenges, we propose OmniHSR, which predicts band-shared spatial operators instead of spectral values (Figure 1b). Every pixel of a low-resolution observation is a spatial mixture of the high-resolution spectra within its footprint, and all bands are mixed in the same way, so a high-resolution spectrum at any position can be approximated by combining the observed spectra around it, and the network only has to decide how they are combined. Such an operator carries no band dimension, so it acts on all bands of any sensor, and the network runs once regardless of the band count. Networks that predict filters for their input already work this way in denoising and video super-resolution (Mildenhall et al., 2018; Jo et al., 2018), for a fixed set of channels on a fixed output grid.

Predicting such operators for any sensor at any factor poses two difficulties. The network needs an input of fixed size while the band count changes with the sensor, and the operators are predicted on the low-resolution grid but must act on a target grid at any factor. Cross-Spectral Mapping (CSM) resamples an input with any number of bands to fixed reference positions and predicts, for every low-resolution pixel, a local operator with a Gaussian support (Kerbl et al., 2023; Hu et al., 2025; Chen et al., 2025a). Continuous Operator-Field Reconstruction (COFR) composes these operators along their supports into a continuous field, evaluates it on the target grid at any factor, and applies it to all original bands.

With the backbone and training shared, predicting operators is more accurate than predicting spectral values on all seven datasets. Trained solely on the 31-band ARAD (Arad et al., 2022) with only 0.538M parameters, OmniHSR outperforms all directly transferred baselines on six unseen datasets without target-domain training data or adaptation, and across twelve factors from $\times 2$ to $\times 48$ it improves the average PSNR on Pavia U (Gamba, 2002) and Chikusei (Yokoya & Iwasaki, 2016) by 0.55 dB over the strongest baseline. It also surpasses the baselines trained from scratch or adapted on the target sensor and runs up to $36\times$ faster on sensors with more than 31 bands (Figure 1d). Our main contributions are as follows.

- We propose OmniHSR, a cross-sensor arbitrary-scale HSR framework that predicts band-shared spatial operators instead of spectral values, so that one model trained once serves unseen sensors at any factor zero-shot.
- We propose CSM, which maps an input with any number of bands to fixed reference positions and predicts local operators with Gaussian supports, and COFR, which composes these operators into a continuous field and applies it to all original bands for arbitrary-scale reconstruction.
- Experiments on seven datasets from six sensors show that, with the backbone and training shared, predicting operators is more accurate than predicting spectral values on every dataset, and that the zero-shot OmniHSR outperforms the baselines whether they are transferred directly, trained from scratch or adapted on the target sensor, and runs up to $36\times$ faster.

2 Related Work

Hyperspectral image super-resolution. Single-image hyperspectral super-resolution has moved from grouped convolutional networks to attention, transformer and state-space designs (Wang et al., 2022; Zhang et al., 2023; Chen et al., 2024b; 2025b). These networks are trained on one sensor at a time, and their last layer has as many channels as that sensor has bands. SQformer, SPG-ASSR and later methods (Jiang et al., 2024; Chen et al., 2024a; Zhang et al., 2025; Wang et al., 2026) extend single-image hyperspectral super-resolution to arbitrary factors. For varying band counts, denoisers that run 3D convolutions or quasi-recurrent units along the spectral axis accept any band count, and models trained on 31-band natural scenes run on sensors with more bands (Wei et al., 2021; Lai et al., 2023). EigenSR (Su et al., 2025) super-resolves the spatial eigenimages of a hyperspectral image with a pretrained RGB network, and MLSR (Zhang et al., 2022) is guided by an RGB image and handles band sets sampled from 31-band datasets, with one model per factor. Remote-sensing foundation models accept different sets of channels through spatial-spectral tokens, wavelength-conditioned embeddings, hypernetworks or attention across channels for recognition and interpretation (Hong et al., 2024; Li et al., 2025; Wang et al., 2025b; Baumann et al., 2026). Fusion methods reach arbitrary factors or band counts with a high-resolution guide image (Chen et al., 2022; Liang et al., 2026; Li et al., 2026; Wang et al., 2025a), and high-pass filtering in pansharpening adds the high-pass detail of a panchromatic image to the resampled multispectral bands (Chavez et al., 1991). OmniHSR needs no wavelength, guide image or adaptation, covers any factor with one model, transfers from 31 to up to 145 bands, and its network outputs no spectral value.

Arbitrary-scale super-resolution and predicted filters. Meta-SR (Hu et al., 2019) and ArbSR (Wang et al., 2021) condition the upsampling layer on the factor, LIIF (Chen et al., 2021) queries a local implicit function at continuous coordinates, and later work refines the query with texture estimation, learned ensembles, latent modulation, neural operators, normalizing flows and diffusion (Lee & Jin, 2022; Cao et al., 2023; He & Jin, 2024; Wei & Zhang, 2023; Yao et al., 2023; Gao et al., 2023; Xu et al., 2025); CUF (Vasconcelos et al., 2023) learns continuous upsampling kernels inside such a network. Gaussian splatting represents a scene or an image with primitives of continuous spatial support (Kerbl et al., 2023; Zhang et al., 2024), and GaussianSR, GSASR and ContinuousSR (Hu et al., 2025; Chen et al., 2025a; Peng et al., 2026) predict 2D Gaussians from the low-resolution image and render them onto any target grid. These methods decode the value at a query position from network features into a number of channels fixed at training time. Predicting a filter and applying it to the input is established in dynamic filtering, burst denoising, video super-resolution and feature upsampling (Jia et al., 2016; Mildenhall et al., 2018; Jo et al., 2018; Wang et al., 2019). In joint image filtering, DKN (Kim et al., 2021) makes each predicted kernel sum to zero, applies it to the bicubic-upsampled target and adds the result back, and it runs separately on each channel of a multi-channel target. LeRF (Li et al., 2023) interpolates the input image with an anisotropic Gaussian predicted for every input pixel, and Wronski et al. (2019) merge a burst by splatting anisotropic kernels onto an arbitrary output grid. OmniHSR keeps the continuous support of the Gaussian primitives and lets every primitive carry one zero-sum operator, which is predicted from the low-resolution image alone and acts on every band of an unseen sensor following the band-shared prior.

3 Proposed Method

Cross-sensor arbitrary-scale HSR reconstructs a high-resolution hyperspectral image $\hat{Y} \in \mathbb{R}^{sh \times sw \times B}$ from a low-resolution observation $X \in \mathbb{R}^{h \times w \times B}$, where the band number B varies across sensors and the scale factor s is arbitrary. OmniHSR aims to build a unified model that generalizes across different band configurations and generates outputs at arbitrary scales.

OmniHSR consists of two modules, Cross-Spectral Mapping (CSM) and Continuous Operator-Field Reconstruction (COFR), and its overall pipeline is illustrated in Figure 2. Specifically, First, CSM resamples X along the band axis to a fixed number M of bands, feeds the result to an encoder and predicts hw 2D Gaussian primitives conditioned on the factor s. Each primitive has a Gaussian spatial support, given by its center μ_n, covariance Σ_n and opacity α_n, and a local spatial operator $\mathbf{k}_n$, which is a small spatial filter,

$$\{X,\, s\} \xrightarrow{\text{CSM}} \left\{(\mu_n, \Sigma_n, \alpha_n, \mathbf{k}_n)\right\}_{n=1}^{hw}. \tag{1}$$

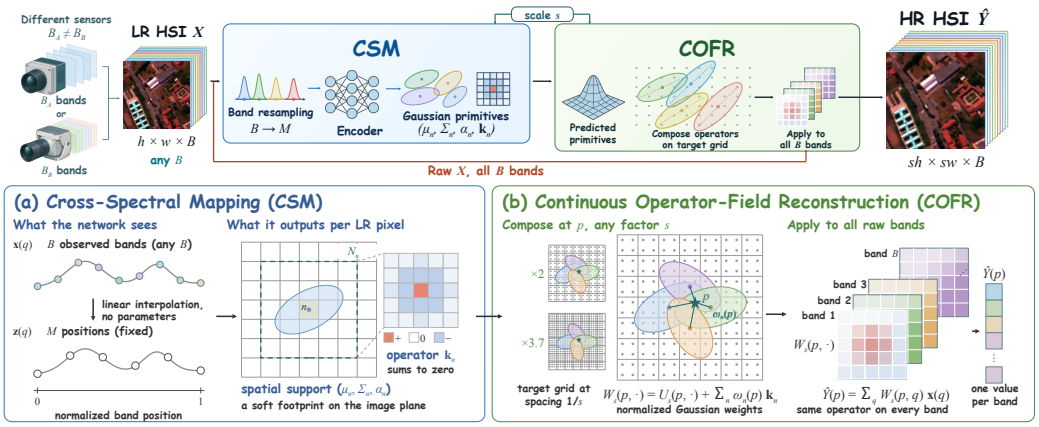


Figure 2: Overview of OmniHSR. (a) CSM resamples B input bands to M fixed normalized band positions and predicts scale-conditioned spatial supports and zero-sum operators. (b) COFR composes the operators with normalized Gaussian weights into one operator at target positions spaced $1/s$ apart and applies it to all raw B bands.

These primitives carry no band axis, and the M bands appear only inside the network. Then, COFR composes the operators of all primitives according to their spatial supports into one reconstruction operator on the $sh \times sw$ target grid and applies it to the raw input X, with all B bands sharing the same coefficients, which gives $\hat{Y}$,

$$\Big(\{(\mu_n, \Sigma_n, \alpha_n, \mathbf{k}_n)\}_{n=1}^{hw}, X,\ s\Big) \xrightarrow{\text{COFR}} \hat{Y}. \tag{2}$$

3.1 Band-Shared Operator Prior

Observation and band-shared operator prior. Let $Y \in \mathbb{R}^{sh \times sw \times B}$ be the high-resolution image, and let the observation X be obtained by applying the same spatial degradation to every band. The bicubic upsampling $U_s X$ combines the observed pixels with fixed weights that sum to one and recovers only the smooth part of Y. Within a local window, we let a zero-sum 5×5 operator $\mathbf{k}$ change how the observed pixels are combined,

$$Y_b \approx U_s\big((\delta + \mathbf{k}) * X_b\big), \qquad \sum_q k_q = 0, \tag{3}$$

where b indexes the bands, $*$ applies the operator of each window on the low-resolution grid and δ is the identity operator, so the combination $\delta + \mathbf{k}$ keeps weights that sum to one and reduces to bicubic interpolation when $\mathbf{k} = 0$. When all bands undergo the same spatial degradation, how the pixels in a window are combined is decided by the same spatial structure, so one $\mathbf{k}$ should serve all bands, which we call the band-shared operator prior.

This prior decides what the network should output. One operator serves all bands, so the network only needs to output one operator without a band dimension for every primitive. This operator can be determined from a fixed set of bands, so the network input only needs a fixed set of band positions. CSM realizes these two points (Section 3.2). The values differ from band to band, and COFR takes them directly from the raw observation while it places the operators on the target grid at any factor (Section 3.3).

3.2 Cross-Spectral Mapping

Input side. Let the B bands of X be ordered by increasing wavelength, with the normalized position of band b written as $(b-1)/(B-1)$. CSM interpolates the spectrum of every pixel linearly along the band axis onto M equally spaced reference positions in $[0, 1]$, which gives the network input Z,

then extracts features with an encoder E_θ and modulates them by the factor s,

$$\mathbf{z}(q) = C_B\,\mathbf{x}(q), \qquad F_s = \gamma(s) \odot E_\theta(Z) + \beta(s), \tag{4}$$

where $\mathbf{x}(q) \in \mathbb{R}^B$ and $\mathbf{z}(q) \in \mathbb{R}^M$ are the spectra of X and Z at low-resolution pixel q, $C_B \in \mathbb{R}^{M\times B}$ is the matrix of this linear interpolation, which depends only on B and has no learnable parameter, and $M = 31$ in this paper; $\gamma(s)$ and $\beta(s)$ are produced from $\log_2 s$ by a two-layer perceptron.

These reference positions record only the order of the bands, so sensors with different spectral ranges share one input axis without wavelength values. Band order is enough because Z serves only to predict spatial operators, which are decided by the spatial structure shared by all bands (Section 3.1).

Output side. From F_s, CSM predicts one 2D Gaussian primitive at every low-resolution pixel, hw primitives in total, and the pixel of primitive n is written as q_n. A primitive is described by a center μ_n, a covariance Σ_n, an opacity α_n and a 5×5 local spatial operator $\mathbf{k}_n$,

$$\big(\mu_n,\ \Sigma_n,\ \alpha_n,\ \mathbf{k}_n\big), \qquad \mu_n \in \mathbb{R}^2, \quad \Sigma_n \in \mathbb{R}^{2\times 2}, \quad \alpha_n \in (0,1), \quad \mathbf{k}_n \in \mathbb{R}^{25}. \tag{5}$$

The first three give the spatial support of the primitive, and $\mathbf{k}_n$ is the operator that this primitive will apply to X. The prediction head outputs 25 coefficients, and their mean is subtracted so that they sum to zero,

$$\sum_{q\in\mathcal{N}_n} k_{n,q} = 0, \tag{6}$$

where $\mathcal{N}_n$ is the 5×5 window of low-resolution pixels centered at q_n and $k_{n,q}$ is the entry of $\mathbf{k}_n$ at q. With the zero sum, the weights with which the observed pixels are combined still sum to one, so a region whose pixels are all equal stays unchanged. No parameter of the network is tied to B, so one set of weights serves any band count.

3.3 Continuous Operator-Field Reconstruction

Eq. (3) places the operator of a window on the target grid with bicubic weights, which are fixed and symmetric around every low-resolution pixel. COFR lets every primitive carry its own operator and places it with the anisotropic Gaussian support of the primitive, which can stretch along the edges that run between low-resolution pixels and can be evaluated at any continuous position.

Composing onto any target grid. All positions are measured in units of low-resolution pixels, so the target grid at factor s has a pixel spacing of $1/s$. At a continuous position p, the response of primitive n is given by an anisotropic Gaussian, and normalizing it by the total response of all primitives gives the weight $\omega_n(p)$,

$$G_n(p) = \exp\Big(-\tfrac{1}{2}\,(p-\mu_n)^\top \Sigma_n^{-1}(p-\mu_n)\Big), \qquad \omega_n(p) = \frac{\alpha_n G_n(p)}{\sum_j \alpha_j G_j(p) + \epsilon}, \tag{7}$$

where ϵ is a small constant.

Continuous operator field. At a continuous position p, mixing the operators of the primitives around p with these weights gives one local operator on the observed pixels,

$$A_s(p,q) = \sum_{n\,:\,q\in\mathcal{N}_n} \omega_n(p)\, k_{n,q}. \tag{8}$$

Each coefficient $A_s(p,q)$ is a scalar that does not depend on the band. In this way, the discrete operators $\mathbf{k}_n$ extend to a continuous operator field A_s over the whole image plane, and the target grid at factor s is one sampling of A_s.

Reconstruction as an operator. COFR adds the operator field to the bicubic weights, which sum to one, and applies the result to all bands of X. With the spatial positions flattened, the reconstruction reads

$$\hat{Y} = W_s X, \qquad W_s = U_s + R_{Z,s} K_{Z,s}, \qquad W_s\mathbf{1} = \mathbf{1}, \tag{9}$$

where X and $\hat{Y}$ are flattened into $hw \times B$ and $s^2hw \times B$ matrices, U_s is the bicubic upsampling matrix at factor s, and $\mathbf{1}$ is the all-ones vector. Row n of $K_{Z,s}$ holds $\mathbf{k}_n$ placed on the window $\mathcal{N}_n$

Table 1: PSNR↑ (dB) at twelve factors on the in-domain ARAD and on two unseen sensors. **Bold**: best; underline: second best; gray: below bicubic interpolation. Time: mean time per image.

Dataset	Method	In-range integers			In-range fractional				Out-of-range extrapolation					Time (ms)
		×2	×4	×8	×2.5	×3.7	×5.3	×7.6	×12	×16	×24	×36	×48	
ARAD (31 bands)	Bicubic	44.04	37.50	33.37	41.60	38.08	35.61	33.70	31.52	30.50	29.20	27.94	27.05	–
	LIIF	44.48	37.96	33.56	42.15	38.56	35.99	33.90	31.47	30.31	28.89	27.55	26.65	16.5
	LTE	45.33	38.38	33.87	42.83	39.01	36.34	34.23	31.79	30.65	29.22	27.79	26.87	17.6
	Meta-SR	45.42	38.33	33.81	42.79	38.94	36.27	34.16	31.73	30.60	29.17	27.75	26.84	51.1
	CiaoSR	45.39	38.39	33.86	42.80	39.00	36.34	34.21	31.75	30.59	29.12	27.66	26.71	57.2
	SRNO	45.58	38.49	33.92	42.94	39.10	36.41	34.28	31.83	30.67	29.26	27.82	26.76	15.6
	GaussianSR	40.81	36.38	32.61	39.46	36.84	34.70	32.91	30.82	29.82	28.55	27.26	26.36	12.5
	OmniHSR	**48.41**	**40.33**	**35.24**	**45.34**	**41.03**	**38.00**	**35.63**	**33.02**	**31.71**	**30.09**	**28.43**	**27.42**	**11.5**
Pavia U (103 bands)	Bicubic	32.64	28.13	25.04	30.92	28.51	26.78	25.32	23.51	22.93	22.41	21.89	21.54	–
	LIIF	32.62	28.11	24.93	30.96	28.52	26.74	25.20	23.39	22.81	22.28	21.75	21.43	115.4
	LTE	33.03	28.34	25.13	31.28	28.75	26.93	25.39	23.47	22.91	22.37	21.84	21.47	120.2
	Meta-SR	33.14	28.35	25.07	31.30	28.76	26.91	25.33	23.45	22.89	22.34	21.80	21.43	357.4
	CiaoSR	33.03	28.35	25.12	31.25	28.75	26.94	25.38	23.48	22.88	22.30	21.71	21.39	398.3
	SRNO	33.24	28.44	25.19	31.37	28.83	27.01	25.45	23.49	22.91	22.37	21.84	21.34	106.9
	GaussianSR	31.84	27.86	24.73	30.46	28.23	26.51	25.00	23.30	22.74	22.22	21.67	21.29	85.0
	OmniHSR	**34.29**	**28.94**	**25.62**	**32.17**	**29.33**	**27.46**	**25.90**	**23.84**	**23.12**	**22.57**	**21.98**	**21.57**	**20.5**
Chikusei (128 bands)	Bicubic	33.04	27.60	24.50	31.17	28.07	26.05	24.70	23.26	22.53	21.78	**21.10**	20.66	–
	LIIF	32.82	27.52	24.38	31.09	28.02	25.98	24.61	23.12	22.38	21.60	20.93	20.53	165.3
	LTE	33.65	27.89	24.58	31.71	28.42	26.26	24.82	23.32	22.55	21.73	21.01	20.57	171.4
	Meta-SR	33.76	27.85	24.54	31.71	28.38	26.21	24.77	23.26	22.50	21.68	20.98	20.56	512.3
	CiaoSR	33.68	27.88	24.58	31.68	28.39	26.24	24.81	23.29	22.52	21.66	20.90	20.50	570.0
	SRNO	33.81	27.96	24.62	31.81	28.48	26.31	24.85	23.35	22.57	21.74	21.03	20.58	152.4
	GaussianSR	31.48	27.12	24.15	30.11	27.53	25.66	24.36	22.94	22.22	21.48	20.77	20.31	121.3
	OmniHSR	**35.44**	**28.95**	**25.20**	**33.13**	**29.51**	**27.10**	**25.47**	**23.78**	**22.93**	**21.92**	**21.10**	**20.67**	**27.1**

and $R_{Z,s}$ collects the weights $\omega_n(p)$; both are predicted from Z and s, and the entries of $R_{Z,s}K_{Z,s}$ are the coefficients $A_s(p,q)$ of Eq. (8). Every row of U_s sums to one and every row of $K_{Z,s}$ sums to zero, so every row of W_s sums to one, and W_s reduces to bicubic interpolation when every operator is zero. Row p of Eq. (9), drawn in Figure 2b, reads $\hat{Y}(p) = \sum_q W_s(p,q)\,\mathbf{x}(q)$ with $W_s(p,q) = U_s(p,q) + A_s(p,q)$, so the network decides only how the observed pixels are combined, and every spectrum of $\hat{Y}$ is an affine combination of the spectra of X.

What the operators add. Applied to X, the operator field adds to the bicubic reconstruction at p

$$\sum_q A_s(p,q)\,\mathbf{x}(q) = \sum_n \omega_n(p)\,\mathbf{r}_n, \qquad \mathbf{r}_n = \sum_{q\in\mathcal{N}_n} k_{n,q}\,\mathbf{x}(q) = \sum_{q\in\mathcal{N}_n} k_{n,q}\big[\,\mathbf{x}(q) - \mathbf{x}(q_n)\,\big], \quad (10)$$

where the first equality exchanges the order of summation in Eq. (8) and the last follows from Eq. (6), so $\mathbf{r}_n$ consists entirely of spectral differences to the center pixel and is computed without any parameter that depends on B.

For a given W_s, the reconstruction commutes with any affine spectral map applied to every pixel. For any $P \in \mathbb{R}^{B'\times B}$ and $\mathbf{c} \in \mathbb{R}^{B'}$,

$$W_s\big(XP^\top + \mathbf{1}\mathbf{c}^\top\big) = \big(W_sX\big)P^\top + \mathbf{1}\mathbf{c}^\top, \quad (11)$$

where the linear part holds because W_s acts only on the spatial axes, and the constant part holds because every row of W_s sums to one. Taking a subset of the bands, merging neighboring bands and applying a per-band gain and offset are all such maps, and for a given W_s they pass through the reconstruction exactly, so the band definition of a sensor reaches the output of OmniHSR only through the prediction of W_s from Z, while it reaches every output value of a network that outputs spectral values. Section 4 tests this prediction on sensors unseen in training. Rows that sum to one are also the partition-of-unity condition of interpolation kernels, which the zero sum keeps for the learned operator field. Computationally, the encoder $E_\theta(Z)$ runs once for any number of requested factors, and the only step whose cost grows with B is the product W_sX, which is linear in B.

Table 2: SSIM↑ and SAM↓ (degrees) on Pavia U at the factors of Table 1. **Bold**: best; underline: second best; gray: worse than bicubic interpolation.

Metric	Method	In-range integers			In-range fractional				Out-of-range extrapolation				
		×2	×4	×8	×2.5	×3.7	×5.3	×7.6	×12	×16	×24	×36	×48
SSIM↑	Bicubic	0.918	0.762	0.604	0.875	0.781	0.692	0.617	0.538	0.516	0.499	0.489	**0.483**
	LIIF	0.917	0.764	0.602	0.876	0.783	0.694	0.616	0.536	0.513	0.497	0.486	0.481
	LTE	0.923	0.774	0.612	0.884	0.793	0.703	0.625	0.541	0.517	0.500	0.488	0.482
	Meta-SR	0.926	0.774	0.610	0.885	0.793	0.702	0.623	0.539	0.516	0.499	0.487	0.481
	CiaoSR	0.924	0.773	0.612	0.883	0.792	0.703	0.625	0.540	0.516	0.498	0.486	0.481
	SRNO	0.927	0.777	0.614	0.886	0.795	0.706	0.628	0.541	0.517	0.499	0.487	0.478
	GaussianSR	0.901	0.752	0.592	0.862	0.770	0.680	0.605	0.532	0.512	0.496	0.486	0.481
	OmniHSR	**0.943**	**0.803**	**0.639**	**0.907**	**0.819**	**0.732**	**0.654**	**0.557**	**0.526**	**0.504**	**0.490**	**0.483**
SAM↓ (deg)	Bicubic	3.45	5.16	7.28	4.00	4.98	5.94	7.03	8.93	9.85	11.00	12.20	13.06
	LIIF	3.53	5.20	7.56	4.02	4.99	6.05	7.30	9.28	10.24	11.43	12.69	13.45
	LTE	3.28	4.93	7.21	3.77	4.74	5.76	6.95	8.97	9.89	11.11	12.37	13.28
	Meta-SR	3.24	4.93	7.26	3.77	4.74	5.77	7.00	9.02	9.98	11.25	12.54	13.43
	CiaoSR	3.27	4.95	7.24	3.80	4.76	5.77	6.98	9.02	10.00	11.29	12.65	13.41
	SRNO	3.22	4.91	7.28	3.75	4.72	5.75	7.01	9.06	10.03	11.22	12.49	13.65
	GaussianSR	4.28	5.71	7.98	4.65	5.53	6.55	7.73	9.60	10.52	11.64	12.86	13.65
	OmniHSR	**3.09**	**4.70**	**6.62**	**3.63**	**4.55**	**5.42**	**6.41**	**8.37**	**9.40**	**10.61**	**11.99**	**12.89**

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets. OmniHSR is trained on the 900 ARAD training images alone, and the other six datasets, CAVE (Yasuma et al., 2010), Harvard (Chakrabarti & Zickler, 2011), Pavia U and Pavia C (Gamba, 2002), Chikusei (Yokoya & Iwasaki, 2016) and Botswana (Ham et al., 2005), serve only for testing. They come from five sensors unseen in training, with 31 to 145 bands spanning the visible to the short-wave infrared.

Baselines. We compare with six arbitrary-scale methods, the factor-conditioned Meta-SR (Hu et al., 2019), the coordinate-query methods LIIF (Chen et al., 2021), LTE (Lee & Jin, 2022), CiaoSR (Cao et al., 2023) and SRNO (Wei & Zhang, 2023), and the Gaussian-based GaussianSR (Hu et al., 2025). They use the ESSAformer encoder (Zhang et al., 2023) of OmniHSR at twice its width or more and are trained on ARAD for 2000 epochs. Their output dimension equals the training band count, so on a new sensor they run on sliding windows of 31 adjacent bands, with no target-domain training. We also train every baseline on the target scene of Pavia U and Chikusei, from scratch or through two 1×1 band adapters on its frozen ARAD body, and test it on tiles that do not overlap the training tiles.

Metrics and protocol. Every low-resolution input is generated by antialiased bicubic downsampling applied identically to all bands, the shared degradation assumed in Section 3.1. We report PSNR and the spectral angle SAM in degrees. Tables 1 and 2 use 256×256 crops of ARAD, Pavia U and Chikusei at twelve factors, of which $\times12$ to $\times48$ lie outside the training range, and Table 4 uses center crops for close-range data and 64×64 tiles for remote-sensing scenes; values are compared within one table only.

Implementation details. OmniHSR uses the ESSAformer encoder with $C{=}32$, has 0.538M parameters and is trained for 2000 epochs with Adam, batch size 2, 256×256 patches and factors drawn uniformly from $[1.5, 8]$, at a learning rate of 2×10^{-4} halved at epochs 600, 1200 and 1600. The loss adds a degradation-consistency term, a spectral-angle term and a spectral-curvature term to a Charbonnier reconstruction term.

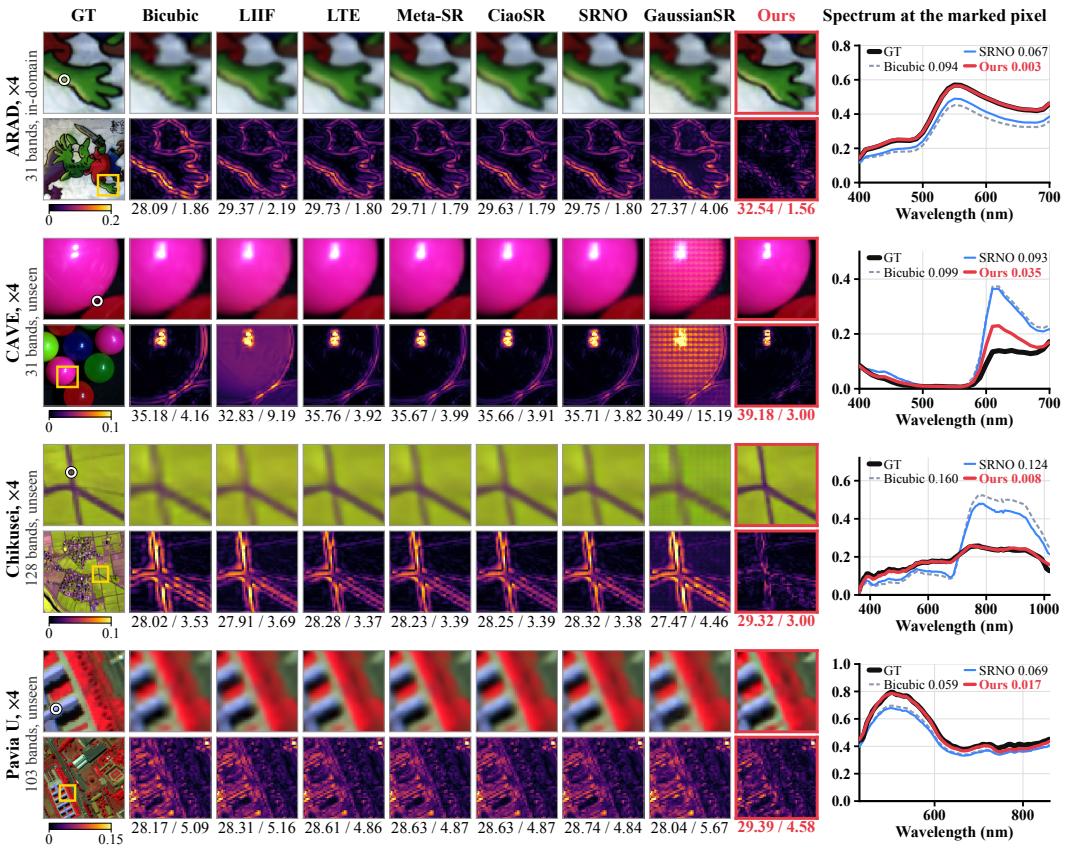


Figure 3: Visual comparison at $\times 4$ on ARAD (in-domain) and on CAVE, Chikusei and Pavia U (unseen sensors, zero-shot). For each dataset, the top row shows a false-color zoom, Chikusei in the near-infrared composite of bands 70, 100 and 36 and Pavia U in that of about 810, 650 and 550 nm, and the bottom row the absolute error averaged over bands; the first column gives the full 256×256 crop with the zoom window. Numbers are PSNR / SAM on the crop. The right column plots the spectrum at the circled pixel with its RMSE in the legend.

4.2 Comparison on Unseen Sensors

Quantitative comparison. As shown in Table 1, with every model trained on ARAD alone, OmniHSR is the best at all twelve factors on the in-domain ARAD and on both unseen sensors. For example, it leads the strongest baseline SRNO by 1.84 dB on ARAD at $\times 4$. On Pavia U and Chikusei it leads the best baseline at each factor by 0.10 to 1.05 dB and by 0.07 to 1.63 dB, and by 0.55 dB on average over the twelve factors of both sensors. All six baselines fall below bicubic interpolation from $\times 36$ on ARAD, $\times 24$ on Chikusei and $\times 12$ on Pavia U, and OmniHSR never does. In Table 2, OmniHSR has the lowest SAM at all twelve factors on Pavia U and the highest SSIM at eleven, level with bicubic interpolation at $\times 48$. It is also the fastest on all three datasets in the time column, and on the 145-band Botswana at $\times 16$ it runs $36.5\times$ faster than CiaoSR (Figure 1d).

Seven datasets. Table 1 tests two unseen sensors at twelve factors, and Table 4 tests one model on many unseen sensors. It runs the same ARAD model on all seven datasets at $\times 4$ against the best of the six baselines in each metric, which run on sliding windows of 31 bands. OmniHSR leads on all seven datasets, by 0.32 to 2.07 dB in PSNR and by 0.10 to 0.40° in SAM, and it is better than bicubic interpolation in both metrics everywhere, while the best baseline falls below bicubic interpolation in PSNR on Botswana and in SAM on Harvard and Botswana.

Training on the target sensor. Training the baselines on the target scene does not close the gap. On the test tiles of Pavia U and Chikusei, the best baseline trained from scratch or through band adapters stays 0.10 to 1.03 dB below the zero-shot OmniHSR at all six factors from $\times 3.5$ to $\times 12$.

Table 3: Ablation at $\times 4$; time and memory on Chikusei. Spectral values: the network outputs the value of every band in place of the operators.

Variant	Pavia U (103)		Chikusei (128)		ms	MiB
	PSNR↑	SAM↓	PSNR↑	SAM↓		
Bicubic	28.13	5.16	27.60	3.95	–	–
Spectral values	27.84	6.86	25.31	8.59	14.8	441
w/o zero-sum	28.82	4.89	28.68	3.55	16.4	524
w/o COFR	28.41	5.09	27.86	3.89	14.2	344
w/o raw bands	28.92	4.75	28.79	3.62	13.9	651
OmniHSR	**28.94**	**4.70**	**28.95**	**3.38**	32.2	716

Table 4: Zero-shot comparison at $\times 4$ with the best of the six baselines in each metric; ARAD is in-domain.

Dataset	PSNR↑ / SAM↓		
	Bicubic	Best baseline	OmniHSR
ARAD (31)	37.50 / 1.74	38.49 / 1.63	**40.33 / 1.38**
CAVE (31)	34.88 / 3.12	35.59 / 2.93	**37.66 / 2.53**
Harvard (31)	43.09 / 2.38	43.53 / 2.41	**44.85 / 2.28**
Pavia U (103)	29.04 / 5.38	29.40 / 5.17	**29.98 / 4.91**
Pavia C (102)	30.05 / 6.39	30.43 / 6.19	**31.11 / 5.97**
Chikusei (128)	29.10 / 11.09	29.73 / 10.93	**30.92 / 10.65**
Botswana (145)	53.07 / 2.38	52.94 / 2.44	**53.26 / 2.34**

Qualitative comparison. Figure 3 compares the reconstructions at $\times 4$ at the level of pixels and spectra. On the in-domain ARAD, the error map of OmniHSR is clearly darker along the outlines. On the unseen CAVE, GaussianSR leaves a grid pattern and the other baselines keep visible errors along the outline of the ball, while the error map of OmniHSR stays dark. At the circled pixel on the edge of the ball, the baselines mix in the spectrum of the neighboring red ball above 600 nm. On the unseen Chikusei, the error maps of the baselines are almost identical to that of bicubic interpolation, and the thin roads between the paddy fields blur into wide bands, which OmniHSR recovers thin and continuous. At the circled road pixel, the baselines mix in the spectra of the neighboring fields, and the spectrum of OmniHSR coincides with the ground truth. On Pavia U, OmniHSR renders the edges of the building rows sharper, and at the circled pixel its spectral RMSE is 0.017 against 0.069 for SRNO.

4.3 Ablation Study

Ablation. Table 3 changes the output of the network and removes the key designs of Section 3 one at a time on the two unseen sensors. Outputting the spectral values outright, with no observed pixels combined, falls below bicubic interpolation on both sensors, by 2.3 dB on Chikusei. Without COFR the operators act once on the low-resolution grid and the result is upsampled bicubically, which costs 0.53 dB on Pavia U and 1.09 dB on Chikusei. Dropping the zero-sum constraint costs 0.12 and 0.27 dB, and applying the operators to 31 resampled bands costs 0.02 and 0.16 dB.

Band sharing. Section 3.1 assumes that one operator serves all bands. A zero-sum 5×5 operator fitted by least squares, with no network, on four bands of every 16×16 window is at least as accurate on the other bands as per-band operators, on held-out pixels of all seven datasets at $\times 2$ and $\times 4$, and one fitted on a single band loses at most 0.01 dB on six of them and up to 0.29 dB on CAVE. Eq. (11) predicts that redefining the bands passes through the reconstruction, and keeping every other band, averaging neighboring pairs or applying a per-band gain or offset before OmniHSR matches applying it afterwards within 0.05 dB on all seven datasets.

5 Conclusion

This paper shows that success in cross-sensor HSR depends on what the network outputs. With the backbone and training shared, a network that outputs spectral values is less accurate than one that predicts band-shared spatial operators on every dataset, even when it serves any band count (Section 4.3), and adapting its interface or training it on the target sensor still leaves it less accurate than a model that has never seen that sensor (Section 4.2). OmniHSR only decides how the observed pixels are combined, transfers from the 31 bands of ARAD to sensors with up to 145 bands without target data (Table 4), and keeps its lead at factors far beyond its training range (Table 1). The spectrum of a new sensor is already in its observation, and what is worth learning is how its pixels are combined in space. One operator shared by all bands assumes that they share their spatial blur and registration, and bands that differ in either call for operators that vary across bands. The low-resolution inputs in this paper are all synthesized, and real or noisy observations remain to be tested.

AI USE STATEMENT

We have not used generative AI tools for any task that requires disclosure under the ICLR 2027 AI Policy for Authors: no such tool was used to develop the method or its conceptual framework, to formulate hypotheses or claims, to design the experiments, to implement the method, to generate or process data, to translate, or to interpret the results. We used generative AI tools only to edit and rephrase the text of this paper for readability. We have reviewed all AI-assisted edits, and we take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses only the public ARAD, CAVE, Harvard, Pavia University, Pavia Centre, Chikusei and Botswana hyperspectral datasets under their original terms of use and involves no new data collection or human subjects. We are not aware of ethical concerns specific to this work.

REPRODUCIBILITY STATEMENT

All experiments use the seven public datasets listed in Appendix A, with the crops, the bicubic degradation and the metrics described in Section 4.1. The same appendix specifies the parameterization of the primitives, the rendering width, the losses and their weights, the operator window, the initialization and the random seed of OmniHSR, and Section 4.1 gives its optimizer, learning-rate schedule and number of epochs. Appendix B.1 specifies how every baseline is used on a new sensor, directly, trained from scratch or through band adapters, and Appendix D gives the protocol of the time measurements. The code will be publicly released.

REFERENCES

Boaz Arad, Radu Timofte, Rony Yahel, Nimrod Morag, Amir Bernat, et al. NTIRE 2022 spectral recovery challenge and data set. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 863–881, 2022.

Alexander Baumann, Leonardo Ayala, Silvia Seidlitz, Jan Sellner, Alexander Studier-Fischer, Berkin Özdemir, Lena Maier-Hein, and Slobodan Ilic. CARL: Camera-agnostic representation learning for spectral image analysis. In *International Conference on Learning Representations (ICLR)*, 2026.

Liheng Bian, Zhen Wang, Yuzhe Zhang, Lianjie Li, Yinuo Zhang, Chen Yang, Wen Fang, Jiajun Zhao, Chunli Zhu, Qinghao Meng, Xuan Peng, and Jun Zhang. A broadband hyperspectral image sensor with high spatio-temporal resolution. *Nature*, 635(8037):73–81, 2024.

Jiezhang Cao, Qin Wang, Yongqin Xian, Yawei Li, Bingbing Ni, Zhiming Pi, Kai Zhang, Yulun Zhang, Radu Timofte, and Luc Van Gool. CiaoSR: Continuous implicit attention-in-attention network for arbitrary-scale image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1796–1807, 2023.

Ayan Chakrabarti and Todd Zickler. Statistics of real-world hyperspectral images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 193–200, 2011.

Pat S. Chavez, Stuart C. Sides, and Jeffrey A. Anderson. Comparison of three different methods to merge multiresolution and multispectral data: Landsat TM and SPOT panchromatic. *Photogrammetric Engineering and Remote Sensing*, 57(3):295–303, 1991.

Du Chen, Liyi Chen, Zhengqiang Zhang, and Lei Zhang. Generalized and efficient 2D Gaussian splatting for arbitrary-scale super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 26435–26445, 2025a.

Guochao Chen, Jiangtao Nie, Wei Wei, Lei Zhang, and Yanning Zhang. Arbitrary-scale hyperspectral image super-resolution from a fusion perspective with spatial priors. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–11, 2024a.

Lihui Chen, Zhibing Lai, Gemine Vivone, Gwanggil Jeon, Jocelyn Chanussot, and Xiaomin Yang. ArbRPN: A bidirectional recurrent pansharpening network for multispectral images with arbitrary numbers of bands. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–18, 2022.

Shi Chen, Lefei Zhang, and Liangpei Zhang. Cross-scope spatial-spectral information aggregation for hyperspectral image super-resolution. *IEEE Transactions on Image Processing*, 33:5878–5891, 2024b.

Shi Chen, Lefei Zhang, and Liangpei Zhang. HSRMamba: Contextual spatial-spectral state space model for single hyperspectral image super-resolution. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 810–818, 2025b.

Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8628–8638, 2021.

Matteo Ciotola, Giuseppe Guarino, Gemine Vivone, Giovanni Poggi, Jocelyn Chanussot, Antonio Plaza, and Giuseppe Scarpa. Hyperspectral pansharpening: Critical review, tools, and future perspectives. *IEEE Geoscience and Remote Sensing Magazine*, 13(1):311–338, 2025.

Paolo Gamba. Hyperspectral remote sensing scenes: Pavia Centre and University. `https://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes`, 2002. ROSIS-03 airborne scenes acquired over Pavia, Italy, in 2002.

Sicheng Gao, Xuhui Liu, Bohan Zeng, Sheng Xu, Yanjing Li, Xiaoyan Luo, Jianzhuang Liu, Xiantong Zhen, and Baochang Zhang. Implicit diffusion models for continuous super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10021–10030, 2023.

Wen-jin Guo, Weiying Xie, Kai Jiang, Yunsong Li, Jie Lei, and Leyuan Fang. Toward stable, interpretable, and lightweight hyperspectral super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22272–22281, 2023.

Jisoo Ham, Yangchi Chen, Melba M. Crawford, and Joydeep Ghosh. Investigation of the random forest framework for classification of hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3):492–501, 2005.

Zongyao He and Zhi Jin. Latent modulated function for computational optimal continuous image representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26026–26035, 2024.

Danfeng Hong, Bing Zhang, Xuyang Li, Yuxuan Li, Chenyu Li, Jing Yao, Naoto Yokoya, Hao Li, Pedram Ghamisi, Xiuping Jia, Antonio Plaza, Paolo Gamba, Jon Atli Benediktsson, and Jocelyn Chanussot. SpectralGPT: Spectral remote sensing foundation model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5227–5244, 2024.

Jintong Hu, Bin Xia, Bin Chen, Wenming Yang, and Lei Zhang. GaussianSR: High fidelity 2D Gaussian splatting for arbitrary-scale image super-resolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 3554–3562, 2025.

Xuecai Hu, Haoyuan Mu, Xiangyu Zhang, Zilei Wang, Tieniu Tan, and Jian Sun. Meta-SR: A magnification-arbitrary network for super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1575–1584, 2019.

Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc Van Gool. Dynamic filter networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Shuguo Jiang, Nanying Li, Meng Xu, Shuyu Zhang, and Sen Jia. SQformer: Spectral-query transformer for hyperspectral image arbitrary-scale super-resolution. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–15, 2024.

Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3224–3232, 2018.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):1–14, 2023.

Beomjun Kim, Jean Ponce, and Bumsub Ham. Deformable kernel networks for joint image filtering. *International Journal of Computer Vision*, 129(2):579–600, 2021.

Zeqiang Lai, Chenggang Yan, and Ying Fu. Hybrid spectral denoising transformer with guided attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13019–13029, 2023.

Jaewon Lee and Kyong Hwan Jin. Local texture estimator for implicit representation function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1929–1938, 2022.

Jiacheng Li, Chang Chen, Wei Huang, Zhiqiang Lang, Fenglong Song, Youliang Yan, and Zhiwei Xiong. Learning steerable function for efficient image resampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Jingtao Li, Yingyi Liu, Xinyu Wang, Yunning Peng, Chen Sun, Shaoyu Wang, Zhendong Sun, Tian Ke, Xiao Jiang, Tangwei Lu, Anran Zhao, and Yanfei Zhong. HyperFree: A channel-adaptive and tuning-free foundation model for hyperspectral remote sensing imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23048–23058, 2025.

Wei Li, Honghui Xu, Yueqian Quan, Zhe Chen, and Jianwei Zheng. Arbitrary-scale spatial-spectral fusion using kernel integral and progressive resampling. *Information Fusion*, 131:104143, 2026.

Yu-Jie Liang, Zihan Cao, Liang-Jian Deng, Yang Yang, and Malu Zhang. Hyperspectral image fusion with spectral-band and fusion-scale agnosticism. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2026.

Ben Mildenhall, Jonathan T. Barron, Jiawen Chen, Dillon Sharlet, Ren Ng, and Robert Carroll. Burst denoising with kernel prediction networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2502–2510, 2018.

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 (ICLR)*, 2026.

Linus Scheibenreif, Michael Mommert, and Damian Borth. Parameter efficient self-supervised geospatial domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27841–27851, 2024.

Xi Su, Xiangfei Shen, Mingyang Wan, Jing Nie, Lihui Chen, Haijun Liu, and Xichuan Zhou. EigenSR: Eigenimage-bridged pre-trained RGB learners for single hyperspectral image super-resolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 7033–7041, 2025.

Romain Thoreau, Valerio Marsocci, and Dawa Derksen. Parameter-efficient adaptation of geospatial foundation models through embedding deflection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9594–9604, 2025.

Cristina N. Vasconcelos, Cengiz Oztireli, Mark Matthews, Milad Hashemi, Kevin Swersky, and Andrea Tagliasacchi. CUF: Continuous upsampling filters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9999–10008, 2023.

Jiaqi Wang, Kai Chen, Rui Xu, Ziwei Liu, Chen Change Loy, and Dahua Lin. CARAFE: Content-aware ReAssembly of FEatures. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3007–3016, 2019.

Longguang Wang, Yingqian Wang, Zaiping Lin, Jungang Yang, Wei An, and Yulan Guo. Learning a single network for scale-arbitrary super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4781–4790, 2021.

Shuo Wang, Mi Wang, Jiaming Wang, and Zhenbei Zhang. DCMArb: Decoupled-collaborative Mamba for arbitrary-scale hyperspectral super-resolution. *ISPRS Journal of Photogrammetry and Remote Sensing*, 239:399–418, 2026.

Ting Wang, Zipei Yan, Jizhou Li, Xile Zhao, Chao Wang, and Michael Ng. Hyperspectral and multispectral image fusion with arbitrary resolution through self-supervised representations. *International Journal of Computer Vision*, 133(11):7515–7535, 2025a.

Xinya Wang, Qian Hu, Junjun Jiang, and Jiayi Ma. A group-based embedding learning and integration network for hyperspectral image super-resolution. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–16, 2022.

Yi Wang, Zhitong Xiong, Chenying Liu, Adam J. Stewart, Thomas Dujardin, Nikolaos Ioannis Bountos, Angelos Zavras, Franziska Gerken, Ioannis Papoutsis, Laura Leal-Taixé, and Xiao Xiang Zhu. Towards a unified Copernicus foundation model for Earth vision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9888–9899, 2025b.

Kaixuan Wei, Ying Fu, and Hua Huang. 3-D quasi-recurrent neural network for hyperspectral image denoising. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):363–375, 2021.

Min Wei and Xuesong Zhang. Super-resolution neural operator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18247–18256, 2023.

Bartlomiej Wronski, Ignacio Garcia-Dorado, Manfred Ernst, Damien Kelly, Michael Krainin, Chia-Kai Liang, Marc Levoy, and Peyman Milanfar. Handheld multi-frame super-resolution. *ACM Transactions on Graphics*, 38(4):1–18, 2019.

Zihao Xu, Yuzhi Tang, Bowen Xu, and Qingquan Li. NeurOp-Diff: Continuous remote sensing image super-resolution via neural operator diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12491–12501, 2025.

Jie-En Yao, Li-Yuan Tsao, Yi-Chen Lo, Roy Tseng, Chia-Che Chang, and Chun-Yi Lee. Local implicit normalizing flow for arbitrary-scale image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1776–1785, 2023.

Fumihito Yasuma, Tomoo Mitsunaga, Daisuke Iso, and Shree K. Nayar. Generalized assorted pixel camera: Postcapture control of resolution, dynamic range, and spectrum. *IEEE Transactions on Image Processing*, 19(9):2241–2253, 2010.

Naoto Yokoya and Akira Iwasaki. Airborne hyperspectral data over Chikusei. Technical Report SAL-2016-05-27, Space Application Laboratory, University of Tokyo, Japan, May 2016.

Mingjin Zhang, Chi Zhang, Qiming Zhang, Jie Guo, Xinbo Gao, and Jing Zhang. ESSAformer: Efficient transformer for hyperspectral image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23073–23084, 2023.

Mingyang Zhang, Shuang Wu, Xiangyu Wang, Maoguo Gong, Fenlong Jiang, Yu Zhou, and Yue Wu. Meta-collaborative learning for arbitrarily scaled hyperspectral image super-resolution. *IEEE Transactions on Geoscience and Remote Sensing*, 63:1–18, 2025.

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 *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 327–345, 2024.

Zhongyang Zhang, Zhiyang Xu, Zia Ahmed, Asif Salekin, and Tauhidur Rahman. Hyperspectral image super-resolution in arbitrary input-output band settings. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, 2022.

APPENDIX

A DATASETS AND IMPLEMENTATION DETAILS

Datasets. Table 5 lists the sensor, platform, band count, spectral range and test size of the seven datasets. Tables 1 and 2 use the center 256×256 crop of each of the 50 ARAD test images, six crops on a 3×2 grid over the 610×340 Pavia U scene and the eight Chikusei test tiles, with a low-resolution side of $\mathrm{round}(256/s)$ at factor s. Every low-resolution input is obtained by antialiased bicubic downsampling, and every model receives the effective factor, the ratio of the high- to the low-resolution side. The seven-dataset evaluation uses center 256×256 crops for close-range data and 64×64 tiles for remote-sensing scenes.

Table 5: Datasets. OmniHSR is trained on ARAD alone; every other dataset is used for testing only, with no target-domain training. Test counts are images for close-range data and non-overlapping 64×64 tiles for remote-sensing scenes. Spectral ranges are those of the sensors; the model receives normalized band positions and no wavelength. Botswana keeps 145 of the 242 Hyperion bands after the removal of water-absorption and noisy bands, so its bands are not evenly spaced in wavelength.

Dataset	Sensor	Platform	Bands	Range (nm)	Test
ARAD (Arad et al., 2022)	Specim IQ	ground	31	400–700	50
CAVE (Yasuma et al., 2010)	CCD with tunable filter	laboratory	31	400–700	31
Harvard (Chakrabarti & Zickler, 2011)	Nuance FX	ground	31	420–720	16
Pavia U (Gamba, 2002)	ROSIS-03	airborne	103	430–860	10 / 45
Pavia C (Gamba, 2002)	ROSIS-03	airborne	102	430–860	187
Chikusei (Yokoya & Iwasaki, 2016)	Headwall Hyperspec-VNIR-C	airborne	128	363–1018	311
Botswana (Ham et al., 2005)	Hyperion (EO-1)	spaceborne	145	400–2500	92

Forward pass. Algorithm 1 lists the forward pass of OmniHSR at a factor s, and the paragraphs below give the details of each step.

Algorithm 1: Forward pass of OmniHSR at a factor s.

Input: $X : (h, w, B)$, factor s
Output: $\hat{Y} : (sh, sw, B)$
$Z \leftarrow C_B X$; // $[h \times w \times M]$, Eq. (4)
$F_s \leftarrow \gamma(s) \odot E_\theta(Z) + \beta(s)$;
$\{(\mu_n, \Sigma_n, \alpha_n, \tilde{\mathbf{k}}_n)\}_{n=1}^{hw} \leftarrow \mathrm{Head}(F_s)$; // one primitive per pixel q_n
$\mathbf{k}_n \leftarrow \tilde{\mathbf{k}}_n - \mathrm{mean}(\tilde{\mathbf{k}}_n)$; // $[hw \times 25]$, Eq. (6)
for $n \in [1, hw]$ **do**
  $\mathbf{r}_n \leftarrow \sum_{q\in\mathcal{N}_n} k_{n,q}\, \mathbf{x}(q)$; // $[B]$, Eq. (10)
$\hat{Y} \leftarrow U_s X$; // bicubic, $[sh \times sw \times B]$
for each target position p, spaced $1/s$ apart **do**
  $\omega_n(p) \leftarrow \alpha_n G_n(p) \,/\, \left(\sum_j \alpha_j G_j(p) + \epsilon\right)$; // Eqs. (7), (12)
  $\hat{Y}(p) \leftarrow \hat{Y}(p) + \sum_n \omega_n(p)\, \mathbf{r}_n$; // all B bands

Primitives. The center of primitive n is its pixel center plus an offset of $0.5\tanh(\cdot)$ low-resolution pixels. The covariance is $\Sigma_n = R(\theta_n)\,\mathrm{diag}(\sigma_{n,1}^2, \sigma_{n,2}^2)\,R(\theta_n)^\top$ with the rotation angle $\theta_n = \pi\tanh(\cdot)$ and the widths $\sigma_{n,i} = \mathrm{softplus}(\cdot) + 0.05$ clipped to $[0.05, 2.5]$ low-resolution pixels, and the opacity is $\alpha_n = \mathrm{sigmoid}(\cdot)$.

Rendering width. The width that each Gaussian uses at rendering carries an analytic anti-aliasing floor that shrinks with the factor and holds no learnable parameter,

$$\sigma_{\mathrm{eff}} = \sqrt{\sigma^2 + (a/s)^2}\,, \tag{12}$$

where σ is the predicted width and $a = 0.5$. Each primitive is rendered in a square window of radius $\lceil 3\sigma_{\max} s \rceil$ high-resolution pixels, where $\sigma_{\max}$ is the largest σ_{eff} in the image, and $\epsilon = 10^{-6}$ in Eq. (7).

Training. The reconstruction term is the Charbonnier loss $\sqrt{(\hat{Y} - Y)^2 + 10^{-6}}$ averaged over pixels and bands. The degradation-consistency term downsamples $\hat{Y}$ to the input size with antialiased bicubic interpolation and applies the same loss against X. The spectral-angle term averages the angle between the predicted and the true spectrum over pixels, with the cosine clipped to $[-1 + 10^{-6}, 1 - 10^{-6}]$, and the spectral-curvature term is the mean absolute difference between the second differences of $\hat{Y}$ and Y along the band axis. The degradation-consistency, spectral-angle and spectral-curvature terms are weighted 0.1, 0.1 and 0.05, and training runs on one RTX 4090D with seed 2026.

Operator window and initialization. The 5×5 window uses replicate padding at the image border, and Appendix C.3 compares other window sizes. The output layer that predicts the operators has zero-initialized weights and bias, so every operator starts at zero and the initial reconstruction operator W_s is bicubic interpolation.

B Comparison with Baselines

B.1 The Three Existing Routes

Figure 4 and Table 6 compare the three ways to use a baseline on a new sensor. They keep the protocol of the target-trained baselines, the 64×64 test tiles of Pavia U and Chikusei that do not overlap their training tiles, with a low-resolution side of $\mathrm{round}(64/s)$. Direct transfer runs the ARAD model on sliding windows of 31 adjacent bands and averages the overlaps, with no target-domain training. Training from scratch follows each method's original recipe on the target scene for 2000 epochs. Adapter tuning freezes the ARAD body, attaches one 1×1 convolution before it and one after it to convert the band count, trains only these two layers from an exact linear band resampling, and runs until the learning rate has annealed.

Figure 4 averages each way over the seven factors from $\times 3.5$ to $\times 16$, taking the best of the six baselines at each factor. Adapter tuning is the strongest of the three ways, and the zero-shot OmniHSR stays above it by 0.21 dB on Pavia U and by 0.60 dB on Chikusei. It also has the highest PSNR of all eighteen configurations at every factor, by 0.03 to 1.03 dB. Direct transfer lifts the stronger methods above bicubic interpolation at small factors, and all six methods fall below it at $\times 12$ on Pavia U and at $\times 16$ on both scenes. Training from scratch leaves all six methods below direct transfer of the same model at $\times 12$ on Chikusei, most for SRNO at 22.87 against 24.60 dB. Adapter tuning lifts every method, and the size of the lift is almost entirely set by the starting point, with a correlation of -0.98 between the two at $\times 4$ on Pavia U and -1.00 on Chikusei. After adapter tuning, LIIF and GaussianSR remain below bicubic interpolation at all seven factors on Chikusei, and only Meta-SR among the six methods is above it at $\times 12$ on Pavia U and at $\times 16$ on both scenes.

The main model can also be run as a 31-band model on sliding windows: under the protocol of Table 1 at $\times 4$, PSNR rises from 28.94 to 29.12 dB on Pavia U and from 28.95 to 29.13 dB on Chikusei, at the cost of 7 to 10 forward passes, 3.6 to 3.8 times the latency, and the requirement of at least 31 bands.

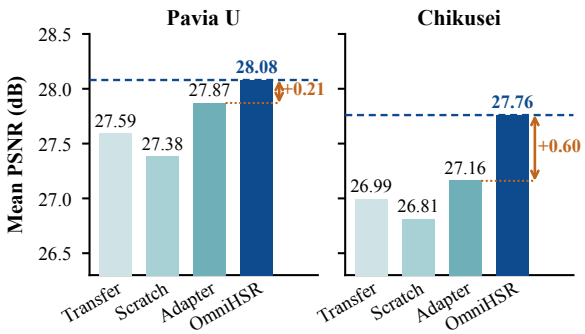


Figure 4: Zero-shot OmniHSR against the best of the six baselines under each way of reusing them on a new sensor, mean PSNR (dB) over seven factors from $\times 3.5$ to $\times 16$ on test tiles of Pavia U and Chikusei. Orange: lead of OmniHSR over the best of the three ways.

Table 6: The three ways to reuse a baseline on a new sensor at seven factors from $\times 3.5$ to $\times 16$ (PSNR↑, dB), the data behind Figure 4. Transfer: the ARAD model run on sliding windows of 31 bands, no target training. Scratch: trained on the target scene for 2000 epochs. Adapter: ARAD body frozen, two 1×1 band adapters tuned on the target scene. Gray: below bicubic interpolation.

Method	Route	Pavia U (103 bands)							Chikusei (128 bands)						
		$\times 3.5$	$\times 4$	$\times 5.5$	$\times 8$	$\times 9.5$	$\times 12$	$\times 16$	$\times 3.5$	$\times 4$	$\times 5.5$	$\times 8$	$\times 9.5$	$\times 12$	$\times 16$
Bicubic		30.62	29.94	28.52	26.89	26.38	25.48	24.96	29.98	29.21	27.67	26.05	25.59	24.60	24.04
LIIF	Transfer	30.23	29.53	28.10	26.50	26.05	25.19	24.70	29.70	28.94	27.42	25.81	25.34	24.34	23.77
	Scratch	29.69	29.09	27.83	26.28	25.88	25.03	24.58	30.50	29.53	27.59	25.73	25.21	23.98	23.36
	Adapter	30.67	29.99	28.54	26.90	26.37	25.36	24.83	29.89	29.14	27.61	25.96	25.47	24.44	23.85
LTE	Transfer	30.86	30.15	28.63	26.82	26.34	25.34	24.80	30.53	29.70	28.03	26.23	25.72	24.62	23.99
	Scratch	30.39	29.71	28.23	26.47	26.02	25.16	24.75	30.60	29.60	27.64	25.63	25.09	23.78	23.07
	Adapter	31.14	30.42	28.89	27.09	26.58	25.46	24.91	30.72	29.91	28.24	26.41	25.89	24.72	24.02
Meta-SR	Transfer	30.83	30.10	28.57	26.80	26.32	25.33	24.80	30.43	29.59	27.92	26.14	25.63	24.57	23.97
	Scratch	30.28	29.59	28.20	26.48	26.02	25.15	24.72	30.78	29.88	27.99	26.01	25.44	24.13	23.44
	Adapter	31.17	30.49	28.96	27.15	26.61	25.52	24.98	30.67	29.85	28.18	26.38	25.87	24.76	24.11
CiaoSR	Transfer	30.80	30.07	28.48	26.63	26.11	25.14	24.77	30.44	29.60	27.90	26.08	25.55	24.46	23.91
	Scratch	30.12	29.50	28.16	26.58	26.15	25.31	24.82	30.56	29.58	27.44	25.22	24.75	23.62	23.02
	Adapter	31.07	30.38	28.78	26.98	26.45	25.36	24.83	30.66	29.87	28.19	26.35	25.81	24.66	24.01
SRNO	Transfer	30.94	30.21	28.65	26.81	26.28	24.98	24.05	30.56	29.73	28.04	26.24	25.72	24.60	23.91
	Scratch	30.66	29.91	28.23	26.31	25.85	24.92	24.48	30.56	29.61	27.57	25.36	24.68	22.87	21.80
	Adapter	31.27	30.56	28.97	27.12	26.59	25.39	24.87	30.78	29.95	28.24	26.41	25.88	24.72	23.93
GaussianSR	Transfer	30.01	29.38	27.97	26.37	25.90	24.92	24.30	29.07	28.40	26.97	25.38	24.90	23.84	23.17
	Scratch	29.72	29.16	27.78	26.15	25.70	24.86	24.48	30.52	29.55	27.54	25.51	24.93	23.69	23.07
	Adapter	30.46	29.84	28.45	26.80	26.25	25.25	24.64	29.59	28.90	27.42	25.78	25.29	24.18	23.48
OmniHSR	zero-shot	**31.53**	**30.83**	**29.19**	**27.41**	**26.96**	**25.62**	**25.01**	**31.81**	**30.92**	**29.03**	**26.92**	**26.33**	**25.01**	**24.30**

B.2 All Baselines on Seven Datasets

Table 7 is the full version of Table 4, with every baseline at $\times 2$, $\times 4$ and $\times 8$. OmniHSR is ahead of every baseline in PSNR and SAM on every dataset at every factor. GaussianSR falls below bicubic interpolation in PSNR on all seven datasets, including the in-domain ARAD, and LIIF on all six unseen datasets at $\times 4$ and $\times 8$. On the 145-band Botswana, which extends to the short-wave infrared, every baseline falls below bicubic interpolation in PSNR at $\times 4$ and $\times 8$, and on Harvard and Botswana every baseline has a higher SAM than bicubic interpolation at all three factors. Paired over the test images and tiles, the PSNR gain of OmniHSR over bicubic interpolation has a 95% bootstrap interval above zero on every dataset at every factor, and at $\times 4$ OmniHSR is above SRNO on every image and tile of all seven datasets.

Table 7: One ARAD-trained model on seven datasets at $\times 2$, $\times 4$ and $\times 8$ (PSNR↑ / SAM↓); the full version of Table 4 with every baseline. Baselines run on sliding windows of 31 bands with no target training; ARAD is in-domain. Remote-sensing scenes are scored on 64×64 test tiles, not on the 256×256 crops of Table 1. Gray: worse than bicubic interpolation. **Bold**: best.

	Method	ARAD (31)	CAVE (31)	Harvard (31)	Pavia U (103)	Pavia C (102)	Chikusei (128)	Botswana (145)
$\times 2$	Bicubic	44.04 / 0.91	40.83 / 1.95	47.47 / 1.90	33.49 / 3.56	34.60 / 4.60	34.62 / 9.46	56.31 / 1.64
	LIIF	44.48 / 1.22	37.03 / 5.56	41.33 / 7.16	33.39 / 3.74	34.61 / 4.83	34.41 / 9.73	50.34 / 5.12
	LTE	45.33 / 0.90	41.44 / 1.90	47.76 / 1.97	34.13 / 3.40	35.27 / 4.44	35.64 / 9.34	56.22 / 1.68
	Meta-SR	45.42 / 0.87	41.51 / 1.87	47.71 / 1.96	34.24 / 3.36	35.38 / 4.40	35.72 / 9.30	56.36 / 1.65
	CiaoSR	45.39 / 0.88	41.27 / 1.91	47.78 / 1.97	34.14 / 3.39	35.29 / 4.44	35.68 / 9.34	56.17 / 1.70
	SRNO	45.58 / 0.87	41.49 / 1.89	47.88 / 1.96	34.34 / 3.34	35.48 / 4.38	35.81 / 9.32	56.22 / 1.69
	GaussianSR	40.81 / 3.30	33.80 / 10.66	38.66 / 9.39	32.58 / 4.48	33.77 / 5.68	32.71 / 10.67	45.94 / 7.00
	OmniHSR	**48.41 / 0.68**	**44.27 / 1.59**	**49.02 / 1.81**	**35.39 / 3.19**	**36.67 / 4.20**	**37.49 / 9.13**	**57.03 / 1.55**
$\times 4$	Bicubic	37.50 / 1.74	34.88 / 3.12	43.09 / 2.38	29.04 / 5.38	30.05 / 6.39	29.10 / 11.09	53.07 / 2.38
	LIIF	37.96 / 1.94	33.45 / 6.29	39.04 / 7.39	28.81 / 5.62	29.84 / 6.62	28.94 / 11.32	48.58 / 5.57
	LTE	38.38 / 1.64	35.52 / 2.93	43.39 / 2.42	29.35 / 5.17	30.41 / 6.19	29.70 / 10.93	52.94 / 2.44
	Meta-SR	38.33 / 1.66	35.41 / 2.97	43.23 / 2.41	29.30 / 5.19	30.34 / 6.21	29.59 / 10.96	52.86 / 2.45
	CiaoSR	38.39 / 1.64	35.54 / 2.94	43.51 / 2.42	29.27 / 5.21	30.33 / 6.23	29.59 / 10.97	52.85 / 2.45
	SRNO	38.49 / 1.63	35.59 / 2.93	43.53 / 2.41	29.40 / 5.20	30.43 / 6.24	29.73 / 10.94	52.81 / 2.46
	GaussianSR	36.38 / 3.63	31.56 / 10.66	37.09 / 9.52	28.65 / 6.03	29.66 / 7.07	28.40 / 11.97	45.67 / 6.97
	OmniHSR	**40.33 / 1.38**	**37.66 / 2.53**	**44.85 / 2.28**	**29.98 / 4.91**	**31.11 / 5.97**	**30.92 / 10.65**	**53.26 / 2.34**
$\times 8$	Bicubic	33.37 / 2.72	30.86 / 4.71	39.56 / 2.76	26.12 / 7.63	27.12 / 8.18	26.00 / 12.73	51.22 / 3.00
	LIIF	33.56 / 2.89	30.16 / 7.58	36.49 / 7.65	25.77 / 8.21	26.77 / 8.61	25.80 / 13.04	47.14 / 6.01
	LTE	33.87 / 2.58	31.37 / 4.40	39.88 / 2.78	26.14 / 7.74	27.18 / 8.14	26.23 / 12.66	51.08 / 3.06
	Meta-SR	33.81 / 2.63	31.23 / 4.47	39.66 / 2.79	26.09 / 7.76	27.10 / 8.17	26.14 / 12.70	50.94 / 3.09
	CiaoSR	33.86 / 2.57	31.39 / 4.41	39.94 / 2.78	25.93 / 7.90	26.97 / 8.32	26.08 / 12.75	51.02 / 3.08
	SRNO	33.92 / 2.58	31.48 / 4.42	39.99 / 2.78	26.01 / 8.00	27.05 / 8.45	26.23 / 12.72	50.44 / 3.13
	GaussianSR	32.61 / 4.28	28.82 / 12.17	34.97 / 9.66	25.64 / 8.42	26.64 / 8.96	25.38 / 13.59	43.90 / 7.68
	OmniHSR	**35.24 / 2.25**	**33.01 / 3.84**	**40.97 / 2.65**	**26.70 / 7.04**	**27.73 / 7.79**	**26.92 / 12.36**	**51.25 / 3.00**

The figures of this appendix choose their crops and zoom windows by fixed rules that look at no reconstruction. Each dataset contributes the 256×256 crop with the median bicubic PSNR at $\times 4$ among its candidates, which are the center crops of the test images for ARAD, CAVE and Harvard, the crops of Table 1 for Pavia U and Chikusei, and non-overlapping crops for Pavia C and Botswana. The zoom window is the one with the strongest gradients of the band-averaged ground truth.

Figure 5 extends Figure 3 to the other three unseen datasets at $\times 4$. OmniHSR has the highest PSNR and the lowest SAM on all three crops, and GaussianSR leaves a grid pattern on Botswana.

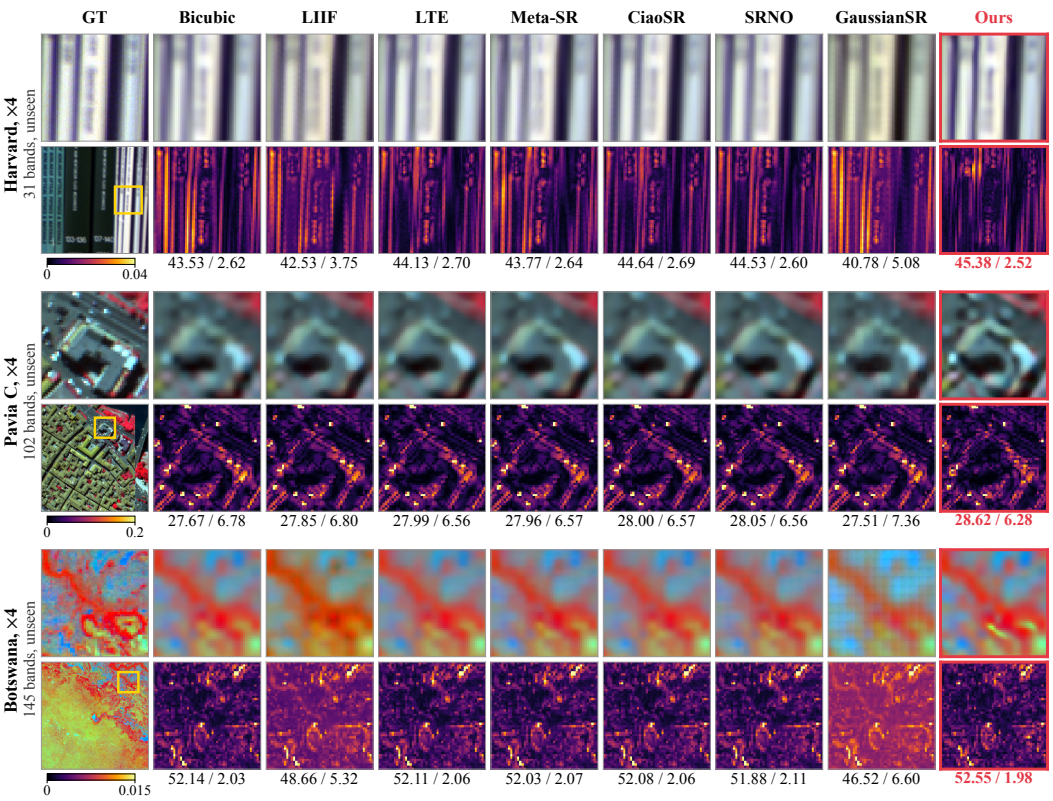


Figure 5: Zero-shot reconstructions at $\times 4$ on the other three unseen datasets, laid out as Figure 3 without the spectra. Pavia C and Botswana use a near-infrared composite of about 810, 650 and 550 nm. Numbers are PSNR / SAM on the crop.

B.3 SSIM AND SAM

Tables 8 and 9 give SSIM and SAM on Chikusei and on the in-domain ARAD under the models, protocol and twelve factors of Table 1, the counterparts of Table 2. On both datasets OmniHSR has the highest SSIM and the lowest SAM at all twelve factors. On Chikusei, every baseline has a higher SAM than bicubic interpolation from $\times 24$ on and a lower SSIM from $\times 36$ on, while OmniHSR stays better than interpolation in both.

Table 10 gives SSIM under the target-trained usages of Appendix B.1, with the better of training from scratch and adapter tuning for each baseline, at $\times 4$, $\times 8$ and $\times 12$, the factors measured for every configuration. The zero-shot OmniHSR has the highest SSIM at $\times 4$ and $\times 8$ on both scenes, and at $\times 12$ all methods and bicubic interpolation lie within 0.02 of each other.

Figure 6 plots the RMSE of every band over the crops of the seven datasets. OmniHSR has the lowest error of all methods in every band on five datasets and in 94% and 84% of the bands on ARAD and Harvard, including the near-infrared bands of Pavia U, Pavia C and Chikusei and the short-wave infrared bands of Botswana, which lie outside the 400 to 700 nm of the training sensor. Taking the 10% of pixels with the strongest gradients of the band-averaged ground truth as edge pixels, among the tenth of edge pixels where bicubic interpolation errs most, the spectrum of OmniHSR is closer to the ground truth than that of SRNO at 76 to 90% of the pixels.

Table 8: SSIM↑ and SAM↓ (degrees) on Chikusei at the factors of Table 1; the Chikusei counterpart of Table 2.

Metric	Method	In-range integers			In-range fractional				Out-of-range extrapolation				
		×2	×4	×8	×2.5	×3.7	×5.3	×7.6	×12	×16	×24	×36	×48
SSIM↑	Bicubic	0.929	0.753	0.587	0.889	0.776	0.668	0.596	0.536	0.515	0.500	0.485	0.477
	LIIF	0.924	0.752	0.587	0.887	0.775	0.669	0.597	0.535	0.514	0.496	0.482	0.473
	LTE	0.936	0.767	0.598	0.900	0.791	0.684	0.608	0.542	0.520	0.500	0.484	0.475
	Meta-SR	0.938	0.766	0.595	0.900	0.789	0.681	0.605	0.540	0.518	0.498	0.482	0.473
	CiaoSR	0.936	0.766	0.597	0.899	0.789	0.683	0.607	0.541	0.519	0.498	0.481	0.472
	SRNO	0.938	0.770	0.599	0.902	0.793	0.686	0.608	0.542	0.520	0.500	0.484	0.473
	GaussianSR	0.899	0.733	0.574	0.861	0.755	0.653	0.583	0.526	0.507	0.492	0.479	0.471
	OmniHSR	**0.957**	**0.813**	**0.631**	**0.928**	**0.834**	**0.731**	**0.643**	**0.558**	**0.529**	**0.503**	**0.487**	**0.479**
SAM↓ (deg)	Bicubic	2.10	3.95	5.79	2.58	3.73	4.76	5.65	6.91	7.64	8.46	9.42	10.15
	LIIF	2.31	4.02	5.90	2.74	3.80	4.81	5.74	7.12	7.90	8.81	9.81	10.57
	LTE	1.95	3.71	5.64	2.39	3.49	4.53	5.48	6.83	7.61	8.53	9.57	10.43
	Meta-SR	1.91	3.73	5.68	2.38	3.51	4.57	5.53	6.90	7.68	8.61	9.65	10.47
	CiaoSR	1.95	3.73	5.65	2.41	3.51	4.55	5.50	6.86	7.63	8.64	9.80	10.60
	SRNO	1.93	3.71	5.65	2.38	3.49	4.54	5.49	6.85	7.64	8.56	9.64	10.45
	GaussianSR	3.31	4.79	6.60	3.69	4.61	5.56	6.45	7.69	8.37	9.19	10.16	10.91
	OmniHSR	**1.70**	**3.38**	**5.29**	**2.14**	**3.17**	**4.18**	**5.10**	**6.49**	**7.26**	**8.27**	**9.41**	**10.09**

Table 9: SSIM↑ and SAM↓ (degrees) on the in-domain ARAD at the factors of Table 1; the ARAD counterpart of Table 2.

Metric	Method	In-range integers			In-range fractional				Out-of-range extrapolation				
		×2	×4	×8	×2.5	×3.7	×5.3	×7.6	×12	×16	×24	×36	×48
SSIM↑	Bicubic	0.978	0.915	0.841	0.963	0.924	0.883	0.847	0.807	0.790	0.768	0.748	0.731
	LIIF	0.981	0.923	0.849	0.967	0.931	0.891	0.855	0.812	0.792	0.769	0.747	0.730
	LTE	0.982	0.926	0.853	0.970	0.934	0.895	0.860	0.817	0.796	0.773	0.749	0.732
	Meta-SR	0.983	0.926	0.853	0.970	0.934	0.895	0.859	0.816	0.796	0.772	0.748	0.730
	CiaoSR	0.983	0.926	0.853	0.970	0.934	0.895	0.859	0.815	0.795	0.770	0.746	0.728
	SRNO	0.983	0.928	0.853	0.970	0.936	0.896	0.859	0.815	0.795	0.772	0.750	0.731
	GaussianSR	0.967	0.907	0.833	0.952	0.915	0.874	0.839	0.800	0.783	0.762	0.742	0.727
	OmniHSR	**0.989**	**0.944**	**0.874**	**0.980**	**0.951**	**0.916**	**0.880**	**0.836**	**0.813**	**0.786**	**0.758**	**0.741**
SAM↓ (deg)	Bicubic	0.91	1.74	2.72	1.15	1.64	2.13	2.62	3.38	3.84	4.52	5.43	6.22
	LIIF	1.22	1.94	2.89	1.42	1.85	2.31	2.80	3.57	4.06	4.79	5.76	6.51
	LTE	0.90	1.64	2.58	1.10	1.54	2.01	2.50	3.24	3.70	4.39	5.31	6.06
	Meta-SR	0.87	1.66	2.63	1.09	1.56	2.04	2.53	3.29	3.77	4.46	5.39	6.11
	CiaoSR	0.88	1.64	2.57	1.10	1.54	2.00	2.48	3.23	3.69	4.39	5.33	6.12
	SRNO	0.87	1.63	2.58	1.09	1.54	2.00	2.49	3.24	3.71	4.40	5.33	6.14
	GaussianSR	3.30	3.63	4.28	3.32	3.58	3.86	4.21	4.83	5.24	5.86	6.68	7.36
	OmniHSR	**0.68**	**1.38**	**2.25**	**0.88**	**1.29**	**1.71**	**2.16**	**2.86**	**3.32**	**3.98**	**4.98**	**5.79**

Table 10: SSIM↑ under the target-trained configurations of Table 6, at the three factors for which every configuration was measured. †: training from scratch was better than adapter tuning. **Bold**: best; underline: second best.

Method	Pavia U (103 bands)			Chikusei (128 bands)		
	×4	×8	×12	×4	×8	×12
Bicubic	0.785	0.641	0.591	0.792	0.648	0.594
LIIF	0.794	0.646	0.590	0.810†	0.661†	0.600†
LTE	0.809	0.655	0.593	0.818	0.672	**0.608**
Meta-SR	0.811	0.656	**0.595**	0.816	0.668	0.605
CiaoSR	0.808	0.651	0.589	0.817	0.670	0.604
SRNO	0.812	0.657	0.593	0.819	0.671	0.603
GaussianSR	0.791	0.642	0.588	0.813†	0.654†	0.593†
OmniHSR	**0.824**	**0.669**	0.594	**0.843**	**0.683**	0.602

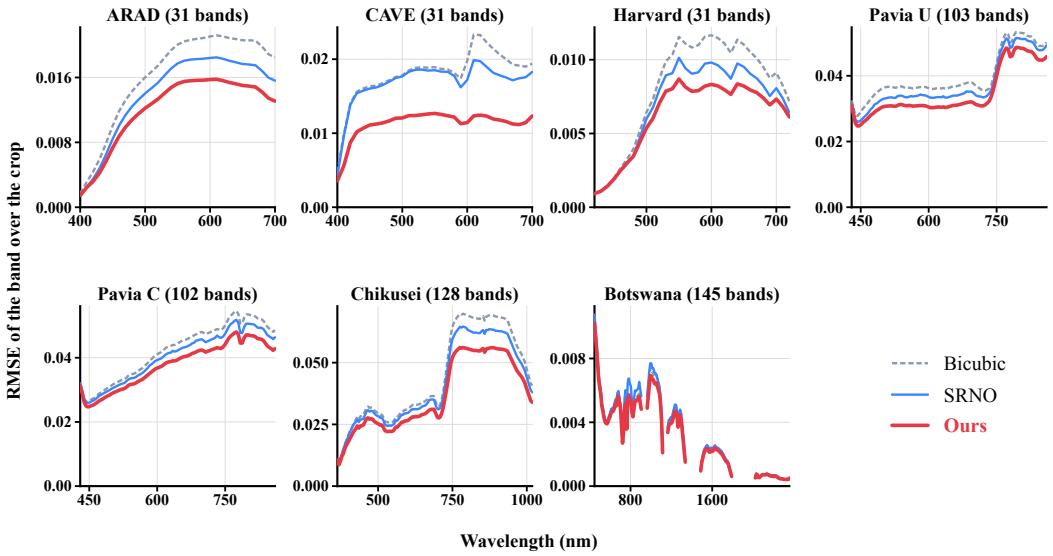


Figure 6: RMSE of every band over the crop at ×4 on the seven datasets, with the crops chosen as in Appendix B.2.

B.4 Reconstructions at $\times 24$

Figure 7 compares the reconstructions at $\times 24$, outside the training range, under the protocol of Table 1, where the input has only 11×11 pixels. The baselines reproduce the blur of bicubic interpolation, while OmniHSR keeps the building rows of Pavia U and the river bank of Chikusei sharper, with the highest PSNR and the lowest SAM on both crops.

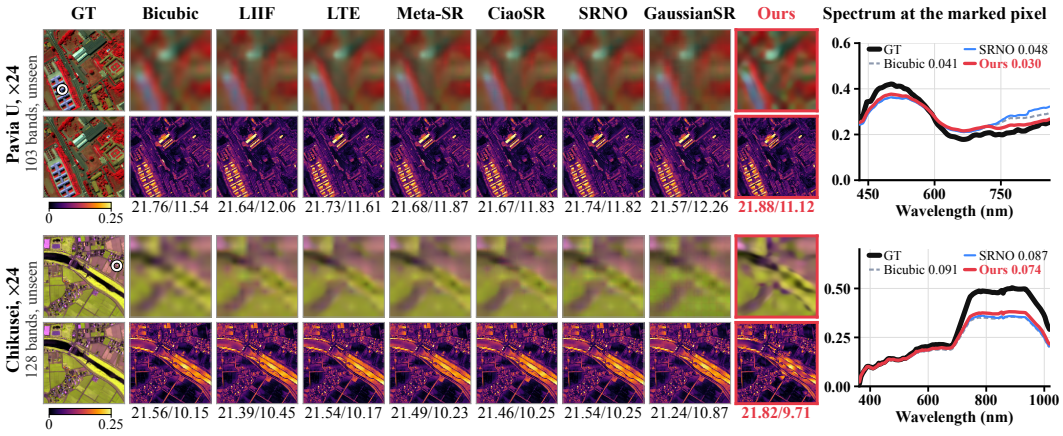


Figure 7: Zero-shot reconstructions at $\times 24$ on Pavia U and Chikusei, laid out as Figure 3. The input has 11×11 pixels, so the zoom shows the whole crop. Numbers are PSNR / SAM on the crop, and the spectrum is taken at the edge pixel with the median bicubic RMSE.

C Analysis of OmniHSR

C.1 Testing the Band-Shared Operator Prior

Table 11 tests the band-shared operator prior of Section 3.1 without any network. The images, crops and degradation are those of the seven-dataset evaluation (Section 4.1 and Table 12). For every image and factor $s \in \{2, 4\}$, we describe Y_b in every window of 16×16 low-resolution pixels by Eq. (3), with a zero-sum 5×5 operator $\mathbf{k}$ of 24 free coefficients. The low-resolution pixels of a window are split into two halves in a checkerboard pattern; $\mathbf{k}$ is fitted by least squares with a small ridge term on the high-resolution pixels of one half and evaluated on the other half, so no high-resolution pixel is used for both.

A shared operator is fitted on a subset of the bands, either one band, four interior bands evenly spaced over the spectrum, or the first half of the spectrum, and is applied unchanged to all bands outside the subset. As the reference, a per-band operator is fitted for every band on the same pixels. The gain of an operator is $10 \log_{10}$ of the ratio between the squared errors of bicubic interpolation and of the operator, where the errors are accumulated over all windows and evaluated bands of an image and the gains are averaged over the images.

An operator fitted on four bands is at least as accurate as the per-band operators on all seven datasets at both factors, and an operator fitted on the first half of the spectrum serves the second half equally well on all datasets except CAVE. These two operators are fitted on more pixels than a per-band operator, which explains their slight advantage. The one-band operator is fitted on as many pixels as a per-band operator, so its column separates the sharing from the number of fitting pixels. It loses at most 0.01 dB on six datasets, while on CAVE it loses 0.29 dB at $\times 2$ and 0.13 dB at $\times 4$, so the operator should be determined from several bands spread over the spectrum, which the fixed band positions of CSM provide. The per-band gains, 0.2 to 1.2 dB, come from a linear operator fitted per window and serve only as the reference for sharing. The operators of OmniHSR form a finer class, one for every primitive placed with a Gaussian support, and this test covers the coarser class of one operator per window.

Table 11: Test of the band-shared operator prior at $\times 2$ and $\times 4$, with no network. Per band: PSNR gain (dB) over bicubic interpolation of the operators fitted on each band. Other columns: gain of one shared operator, fitted on one band, four bands or the first half of the spectrum and applied to the remaining bands, minus that of the per-band operators on the same bands. Gray: the shared operator is less accurate.

	$\times 2$				$\times 4$			
Dataset	Per band	1 band	4 bands	First half	Per band	1 band	4 bands	First half
ARAD (31)	1.19	−0.01	+0.03	+0.01	0.49	+0.01	+0.02	+0.01
CAVE (31)	0.73	−0.29	+0.07	−0.10	0.33	−0.13	+0.04	−0.06
Harvard (31)	0.59	+0.06	+0.10	+0.10	0.57	+0.01	+0.03	+0.03
Pavia U (103)	0.64	+0.04	+0.17	+0.05	0.31	+0.04	+0.08	+0.04
Pavia C (102)	0.67	+0.04	+0.14	+0.08	0.31	+0.01	+0.06	+0.03
Chikusei (128)	0.82	+0.00	+0.09	+0.06	0.34	+0.02	+0.06	+0.03
Botswana (145)	0.37	+0.01	+0.12	+0.12	0.17	+0.01	+0.04	+0.04

C.2 Output Path and Operator Constraint

Table 12 compares outputs of the network at $\times 2$, $\times 4$ and $\times 8$ on all seven datasets, with the backbone and training shared. The spectral-value column outputs the spectral values outright and combines no observed pixels, and it falls below bicubic interpolation in PSNR in 19 of the 21 rows and below OmniHSR in all 21. The free-kernel column drops the zero-sum constraint of the operator, the counterpart of the w/o zero-sum row of Table 3. It stays above bicubic interpolation in every row and below OmniHSR in PSNR in 20 of the 21 rows.

Table 12: Output path and operator constraint at $\times 2$, $\times 4$ and $\times 8$ on seven datasets (PSNR↑ / SAM↓). Every model, OmniHSR included, is trained on the 900 ARAD images with the same schedule, shorter than that of the main model, so the OmniHSR column differs from the main model in Table 7. Remote-sensing scenes are scored on 64×64 test tiles, not on the 256×256 crops of Table 1. Gray: worse than bicubic interpolation. **Bold**: best in the row.

	Dataset	Bicubic	Predicted spectral values	Free kernel	OmniHSR
$\times 2$	ARAD (31)	44.04 / 0.91	43.98 / 1.94	47.85 / 0.71	**47.94 / 0.69**
	CAVE (31)	40.83 / 1.95	35.07 / 11.63	43.81 / 1.64	**44.00 / 1.62**
	Harvard (31)	47.47 / 1.90	40.04 / 8.61	48.84 / 1.82	**48.91 / 1.81**
	Pavia U (103)	33.49 / 3.56	32.50 / 5.79	35.26 / 3.26	**35.28 / 3.21**
	Pavia C (102)	34.60 / 4.60	34.03 / 6.99	36.50 / 4.25	**36.55 / 4.21**
	Chikusei (128)	34.62 / 9.46	29.26 / 14.60	37.07 / 9.19	**37.23 / 9.16**
	Botswana (145)	56.31 / 1.64	40.06 / 15.20	56.82 / 1.56	**57.04 / 1.54**
$\times 4$	ARAD (31)	37.50 / 1.74	38.22 / 2.42	39.86 / 1.45	**39.94 / 1.42**
	CAVE (31)	34.88 / 3.12	32.30 / 11.96	37.22 / 2.61	**37.37 / 2.58**
	Harvard (31)	43.09 / 2.38	38.20 / 8.79	44.54 / **2.29**	**44.62 / 2.29**
	Pavia U (103)	29.04 / 5.38	28.66 / 7.05	29.87 / 5.09	**29.92 / 5.02**
	Pavia C (102)	30.05 / 6.39	29.88 / 8.16	30.95 / 6.09	**31.04 / 6.05**
	Chikusei (128)	29.10 / 11.09	26.39 / 15.42	30.53 / 10.78	**30.69 / 10.73**
	Botswana (145)	53.07 / 2.38	39.41 / 15.26	53.17 / 2.35	**53.32 / 2.32**
$\times 8$	ARAD (31)	33.37 / 2.72	34.04 / 3.13	34.88 / 2.35	**34.96 / 2.33**
	CAVE (31)	30.86 / 4.71	29.62 / 12.53	32.59 / 3.97	**32.68 / 3.96**
	Harvard (31)	39.56 / 2.76	35.83 / 8.98	**40.74 / 2.67**	40.68 / **2.67**
	Pavia U (103)	26.12 / 7.63	25.89 / 8.90	26.66 / 7.19	**26.69 / 7.14**
	Pavia C (102)	27.12 / 8.18	26.91 / 9.65	27.61 / 7.92	**27.66 / 7.89**
	Chikusei (128)	26.00 / 12.73	24.27 / 16.49	26.71 / 12.48	**26.74 / 12.47**
	Botswana (145)	51.22 / 3.00	38.64 / 15.44	51.23 / 2.99	**51.29 / 2.98**

Figure 8 compares outputting the spectral values with the operators, both from Table 12, on three unseen datasets at $\times 4$. Outputting the spectral values outright shifts the colors of whole regions on CAVE and fails on the 145-band Botswana, where its error exceeds the color scale everywhere, and its error is above that of the operators in every band of all three crops.

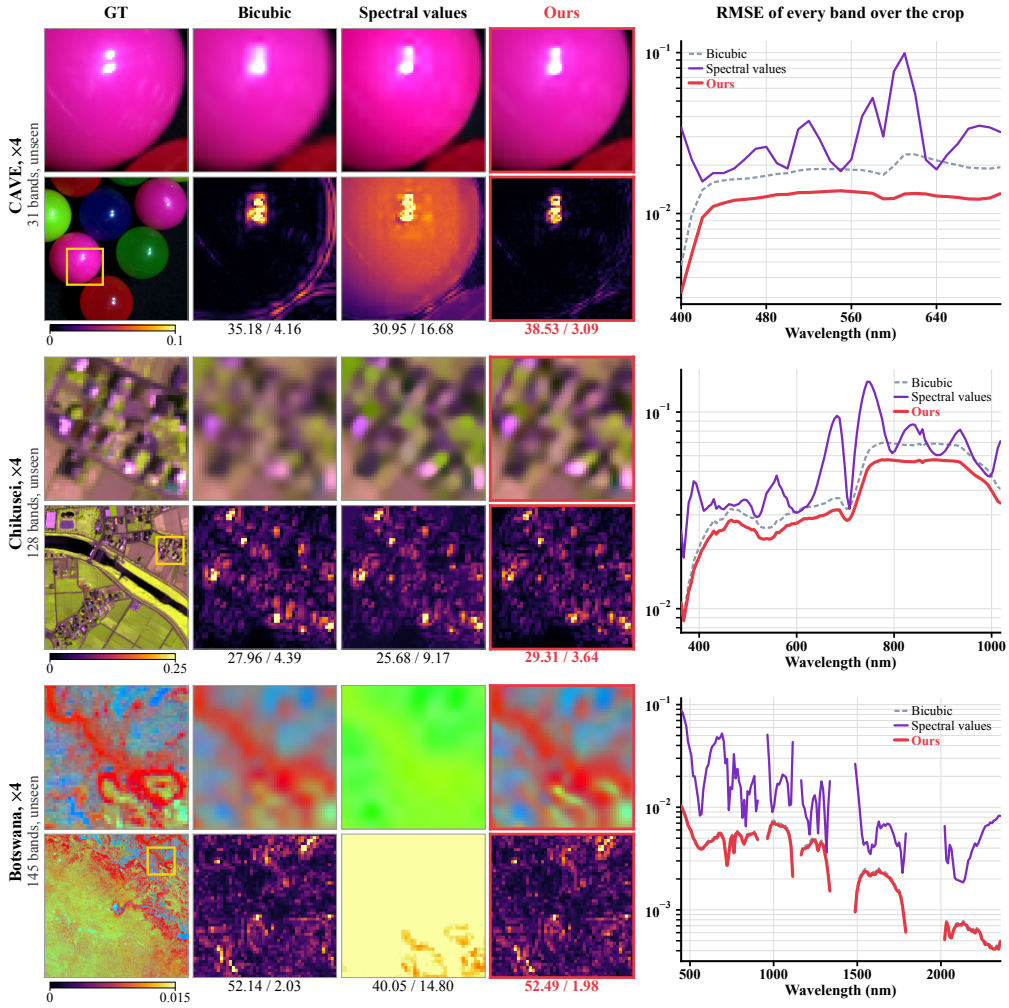


Figure 8: Output paths at $\times 4$ with the models of Table 12. Spectral values: the network outputs the value of every band. The right column gives the RMSE of every band over the crop on a log scale. Numbers are PSNR / SAM on the crop.

C.3 Reference Positions and Operator Window

Table 13 varies the number M of reference positions from 1 to 62. Each model is retrained on the 900 ARAD images and evaluated on the 256×256 crops of Table 1. From $M{=}8$ on, including $M{=}62$, the PSNR on all three datasets stays within 0.1 dB of $M{=}31$. With a single position, ARAD and Chikusei lose 0.41 and 0.55 dB at $\times 4$ and remain above bicubic interpolation. Time and peak memory show no monotonic trend in M.

Table 14 varies the operator window from 3×3 to 9×9 with the same budget, and its 5×5 row is the model in the $M{=}31$ row of Table 13. In PSNR, the 7×7 window stays within 0.04 dB of 5×5 on all three datasets. The 3×3 window is 0.10 to 0.25 dB below 5×5 and keeps 72 to 82% of the gain of 5×5 over bicubic interpolation on the two unseen sensors and about 90% on ARAD. The 9×9 window is 0.01 to 0.09 dB below 5×5. The window changes the parameter count by at most 0.4%, and no larger window is slower than 5×5.

Table 13: Number M of reference positions (PSNR↑ / SAM↓). Every row is trained on the 900 ARAD images with the same schedule, shorter than that of the main model, so the M=31 row, the setting of the main model, differs from Table 1. Time and peak memory on Chikusei.

M	ARAD (31 bands)	Pavia U (103 bands)	Chikusei (128 bands)	Time (ms)	Memory (MiB)
×4					
Bicubic	37.50 / 1.74	28.13 / 5.16	27.60 / 3.95	–	–
1	39.53 / 1.52	28.86 / 4.91	28.23 / 3.81	15.4	490
2	39.76 / 1.46	28.82 / 4.93	28.61 / 3.52	18.7	524
4	39.83 / 1.45	28.87 / 4.83	28.72 / 3.45	23.2	596
8	39.89 / 1.42	28.92 / 4.78	28.84 / 3.44	23.1	595
16	39.86 / 1.42	28.89 / 4.79	28.80 / 3.46	21.8	559
31	39.94 / 1.42	28.91 / 4.81	28.78 / 3.48	21.8	559
62	39.91 / 1.42	28.92 / 4.79	28.80 / 3.46	20.9	556
×8					
Bicubic	33.37 / 2.72	25.04 / 7.28	24.50 / 5.79	–	–
1	34.72 / 2.45	25.65 / 6.76	24.88 / 5.65	14.0	414
2	34.84 / 2.38	25.54 / 7.00	25.03 / 5.44	14.1	425
4	34.84 / 2.35	25.61 / 6.81	25.08 / 5.37	14.2	425
8	34.91 / 2.33	25.58 / 6.76	25.13 / 5.39	19.2	497
16	34.91 / 2.32	25.59 / 6.78	25.11 / 5.43	15.2	467
31	34.96 / 2.33	25.58 / 6.81	25.09 / 5.42	15.1	468
62	34.94 / 2.32	25.59 / 6.80	25.10 / 5.41	14.0	426

Table 14: Operator window (PSNR↑ / SAM↓). Every row is trained on the 900 ARAD images with the same schedule, shorter than that of the main model, so the 5×5 row, the setting of the main model, differs from Table 1. Time and peak memory on Chikusei.

Window	ARAD (31 bands)	Pavia U (103 bands)	Chikusei (128 bands)	Time (ms)	Memory (MiB)
×4					
Bicubic	37.50 / 1.74	28.13 / 5.16	27.60 / 3.95	–	–
3×3	39.70 / 1.49	28.69 / 5.00	28.53 / 3.56	32.1	684
5×5	39.94 / 1.42	28.91 / 4.81	28.78 / 3.48	21.8	559
7×7	39.95 / 1.42	28.92 / 4.78	28.82 / 3.44	21.2	605
9×9	39.85 / 1.44	28.82 / 4.90	28.73 / 3.52	21.8	666
×8					
Bicubic	33.37 / 2.72	25.04 / 7.28	24.50 / 5.79	–	–
3×3	34.81 / 2.37	25.48 / 7.05	24.98 / 5.48	18.4	489
5×5	34.96 / 2.33	25.58 / 6.81	25.09 / 5.42	15.1	468
7×7	34.94 / 2.31	25.58 / 6.80	25.10 / 5.39	14.9	451
9×9	34.87 / 2.33	25.57 / 6.93	25.06 / 5.48	14.2	441

C.4 Training on a Different Source Sensor

Table 15 swaps the training sensor for the 128-band Chikusei, keeps everything else fixed, and places this model next to the ARAD-trained main model on the seven datasets at $\times2$, $\times4$ and $\times8$. On the three 31-band datasets, PSNR and SAM are above bicubic interpolation at $\times2$, $\times4$ and $\times8$ throughout, and so are the remaining datasets except Botswana at $\times8$, where the model ties with interpolation. Transfer holds in both directions, and 31 is only the choice made for the main model of this paper. The Chikusei-trained model gains less than the ARAD-trained one; its training data is 234 tiles of a single scene.

Table 15: Training sensor swapped: the same framework trained on the 128-band Chikusei (234 tiles, 2000 epochs) against the main model trained on the 31-band ARAD (900 images, 2000 epochs), PSNR↑ / SAM↓ on the seven datasets at $\times 2$, $\times 4$ and $\times 8$, single forward pass, no target-domain data for either. ARAD is in-domain for the ARAD-trained model and Chikusei for the Chikusei-trained one. Gray: worse than bicubic interpolation.

	Dataset	Bicubic	OmniHSR trained on ARAD (31 bands)	OmniHSR trained on Chikusei (128 bands)
$\times 2$	ARAD (31)	44.04 / 0.91	48.41 / 0.68	46.59 / 0.79
	CAVE (31)	40.83 / 1.95	44.27 / 1.59	43.12 / 1.76
	Harvard (31)	47.47 / 1.90	49.02 / 1.81	48.44 / 1.83
	Pavia U (103)	33.49 / 3.56	35.39 / 3.19	35.11 / 3.25
	Pavia C (102)	34.60 / 4.60	36.67 / 4.20	36.33 / 4.30
	Chikusei (128)	34.62 / 9.46	37.49 / 9.13	37.68 / 9.01
	Botswana (145)	56.31 / 1.64	57.03 / 1.55	57.02 / 1.55
$\times 4$	ARAD (31)	37.50 / 1.74	40.33 / 1.38	38.99 / 1.60
	CAVE (31)	34.88 / 3.12	37.66 / 2.53	36.33 / 2.87
	Harvard (31)	43.09 / 2.38	44.85 / 2.28	43.93 / 2.32
	Pavia U (103)	29.04 / 5.38	29.98 / 4.91	29.72 / 5.17
	Pavia C (102)	30.05 / 6.39	31.11 / 5.97	30.82 / 6.20
	Chikusei (128)	29.10 / 11.09	30.92 / 10.65	31.39 / 10.37
	Botswana (145)	53.07 / 2.38	53.26 / 2.34	53.22 / 2.35
$\times 8$	ARAD (31)	33.37 / 2.72	35.24 / 2.25	34.33 / 2.59
	CAVE (31)	30.86 / 4.71	33.01 / 3.84	31.82 / 4.38
	Harvard (31)	39.56 / 2.76	40.97 / 2.65	40.10 / 2.73
	Pavia U (103)	26.12 / 7.63	26.70 / 7.04	26.39 / 7.44
	Pavia C (102)	27.12 / 8.18	27.73 / 7.79	27.44 / 8.07
	Chikusei (128)	26.00 / 12.73	26.92 / 12.36	27.53 / 11.95
	Botswana (145)	51.22 / 3.00	51.25 / 3.00	51.23 / 3.01

C.5 Redefining the Bands

Eq. (11) holds for a given W_s, and the band definition of a sensor reaches the output of OmniHSR only through the prediction of W_s from Z (Section 3.3). Table 16 redefines the bands of every test image of the seven-dataset evaluation in four ways and runs OmniHSR on the redefined input TX. It compares the result $f(TX)$ with $Tf(X)$, the reconstruction of the original bands redefined afterwards, which Eq. (11) makes equal to $f(TX)$ if the predicted W_s does not change. The relative change $r = \|f(TX) - Tf(X)\| / \|Tf(X) - U_s TX\|$ measures how much of the contribution of the operator field moves, and Δ compares the two against the redefined ground truth TY, without clipping. At $\times 4$, the median r is no more than 0.09 for subsets and merges and 0.17 for gains and offsets, and at $\times 2$, $\times 4$ and $\times 8$ the PSNR changes by at most 0.05 dB for every transform and dataset. Redefining the bands changes the predicted operators little and leaves the accuracy of the reconstruction unchanged.

Table 16: Redefining the input bands at $\times 4$. Subset: every other band; merge: means of neighboring pairs; gain and offset: a random per-band gain in $[0.8, 1.25]$ or offset in $[0, 0.05]$. r: relative change of the contribution of the operator field, median over images; Δ: PSNR of $f(TX)$ minus that of $Tf(X)$, both against TY (dB).

Dataset	Subset		Merge		Gain		Offset	
	r	Δ	r	Δ	r	Δ	r	Δ
ARAD (31)	0.008	0.00	0.049	−0.01	0.060	0.00	0.089	−0.02
CAVE (31)	0.021	+0.01	0.064	−0.01	0.063	0.00	0.071	+0.01
Harvard (31)	0.060	0.00	0.090	+0.01	0.064	0.00	0.104	−0.03
Pavia U (103)	0.007	0.00	0.038	+0.01	0.077	+0.02	0.160	+0.04
Pavia C (102)	0.017	0.00	0.038	+0.01	0.144	0.00	0.136	−0.03
Chikusei (128)	0.013	0.00	0.025	0.00	0.076	−0.03	0.053	0.00
Botswana (145)	0.018	0.00	0.037	0.00	0.108	−0.01	0.160	0.00

C.6 Blur That Grows Along the Spectrum

Every other experiment degrades all bands identically, the setting that the band-shared prior assumes. Table 17 lets the blur grow along the spectrum. Before the antialiased bicubic downsampling, band b is blurred by a Gaussian of standard deviation $\kappa s(b-1)/(B-1)$ high-resolution pixels, so at $\kappa = 0.5$ the last band carries an extra blur of half a low-resolution pixel, and the models are the ARAD-trained ones of Table 4. Every method loses accuracy as κ grows. At $\kappa = 0.5$, OmniHSR is above SRNO on every image and tile of the seven datasets, by 0.49 to 1.78 dB on average, and above bicubic interpolation on every dataset, with the lowest SAM of the three methods; SRNO stays below bicubic interpolation on Botswana.

Table 17: Blur that grows along the spectrum, PSNR↑ (dB) at $\times 4$. Band b is blurred by a Gaussian of standard deviation $\kappa s(b-1)/(B-1)$ high-resolution pixels before the antialiased bicubic downsampling; $\kappa = 0$ is the setting of Table 4. Gray: below bicubic interpolation. **Bold**: best for each κ.

Dataset	$\kappa = 0$			$\kappa = 0.25$			$\kappa = 0.5$		
	Bicubic	SRNO	OmniHSR	Bicubic	SRNO	OmniHSR	Bicubic	SRNO	OmniHSR
ARAD (31)	37.50	38.49	**40.33**	37.34	38.27	**40.16**	36.85	37.57	**39.18**
CAVE (31)	34.88	35.59	**37.66**	34.72	35.38	**37.51**	34.21	34.73	**36.51**
Harvard (31)	43.09	43.53	**44.85**	42.94	43.37	**44.72**	42.49	42.85	**43.99**
Pavia U (103)	29.04	29.40	**29.98**	28.92	29.28	**29.94**	28.57	28.90	**29.56**
Pavia C (102)	30.05	30.43	**31.11**	29.93	30.30	**31.08**	29.56	29.88	**30.63**
Chikusei (128)	29.10	29.73	**30.92**	28.98	29.58	**30.88**	28.62	29.13	**30.28**
Botswana (145)	53.07	52.81	**53.26**	52.98	52.74	**53.22**	52.75	52.54	**53.03**

C.7 Test-Time Degradations

Table 18 changes only how the test inputs are generated, on the crops of Table 1. Noise adds i.i.d. Gaussian noise with standard deviation σ to the bicubic input. Blur applies an isotropic Gaussian with a standard deviation of $0.25s$, $0.40s$ or $0.55s$ high-resolution pixels and keeps every s-th pixel, and area averages $s \times s$ blocks.

C.8 What the Network Predicts

Figure 9 shows the primitives and operators that OmniHSR predicts at $\times 4$ on crops of ARAD and Chikusei chosen as in Appendix B.2. At edges the supports stretch along the edge. We call a primitive an edge primitive when the gradient magnitude of the band-averaged ground truth at its low-resolution pixel is in the top 10%, and a flat primitive when it is in the bottom half. Among the edge primitives with an axis ratio of at least 1.2, the long axis lies within 30° of the edge direction for 98.5% on ARAD and 94.8% on Chikusei. The median axis ratio is 3.9 at edges and 1.3 in flat

Table 18: PSNR↑ (dB) under test-time degradations on the crops of Table 1. For each dataset: bicubic interpolation, the best of the six baselines, and OmniHSR. Gray: below bicubic interpolation; **bold**: best.

Test input	ARAD (31)			Pavia U (103)			Chikusei (128)		
	Bicubic	Best	OmniHSR	Bicubic	Best	OmniHSR	Bicubic	Best	OmniHSR
×4									
Bicubic (as in training)	37.50	38.49	**40.36**	28.13	28.44	**28.92**	27.60	27.96	**28.94**
Noise σ=0.01	35.07	36.64	**36.75**	27.92	28.23	**28.30**	27.31	27.72	**27.82**
Noise σ=0.02	32.65	**35.26**	35.25	27.34	**27.91**	27.87	26.65	**27.28**	27.26
Noise σ=0.03	30.64	**34.44**	30.78	26.55	**27.69**	26.64	25.85	**27.05**	25.92
Blur $0.25s$	36.83	37.77	**37.79**	27.59	**27.84**	27.82	27.14	**27.46**	27.45
Blur $0.40s$	36.34	36.97	**38.06**	27.31	27.49	**27.97**	26.87	27.12	**27.83**
Blur $0.55s$	35.44	35.79	**36.58**	26.63	26.71	**27.10**	26.16	26.30	**26.90**
Area	37.53	38.76	**39.35**	28.14	28.48	**28.49**	27.59	27.98	**28.36**
×8									
Bicubic (as in training)	33.37	33.92	**35.26**	25.04	25.19	**25.59**	24.50	24.62	**25.16**
Noise σ=0.01	31.98	32.78	**32.91**	24.94	25.02	**25.08**	24.34	24.49	**24.57**
Noise σ=0.02	30.40	31.98	**31.99**	24.63	**24.79**	24.76	23.97	**24.25**	24.20
Noise σ=0.03	28.95	**31.51**	29.07	24.17	**24.65**	24.22	23.47	**24.11**	23.55
Blur $0.25s$	33.27	**33.97**	**33.97**	24.99	**25.15**	25.12	24.36	**24.55**	24.51
Blur $0.40s$	32.91	33.37	**34.52**	24.68	24.82	**25.33**	24.23	24.34	**24.90**
Blur $0.55s$	32.21	32.47	**33.23**	24.11	24.22	**24.59**	23.77	23.84	**24.23**
Area	33.37	34.00	**34.48**	25.05	25.20	**25.36**	24.48	24.60	**24.76**

regions on ARAD, and 3.8 and 1.6 on Chikusei. The operators concentrate on edges as well, with a median norm of 3.5 at edges and 0.9 in flat regions on ARAD and 2.8 and 1.4 on Chikusei, so the correction to bicubic interpolation is small away from edges.

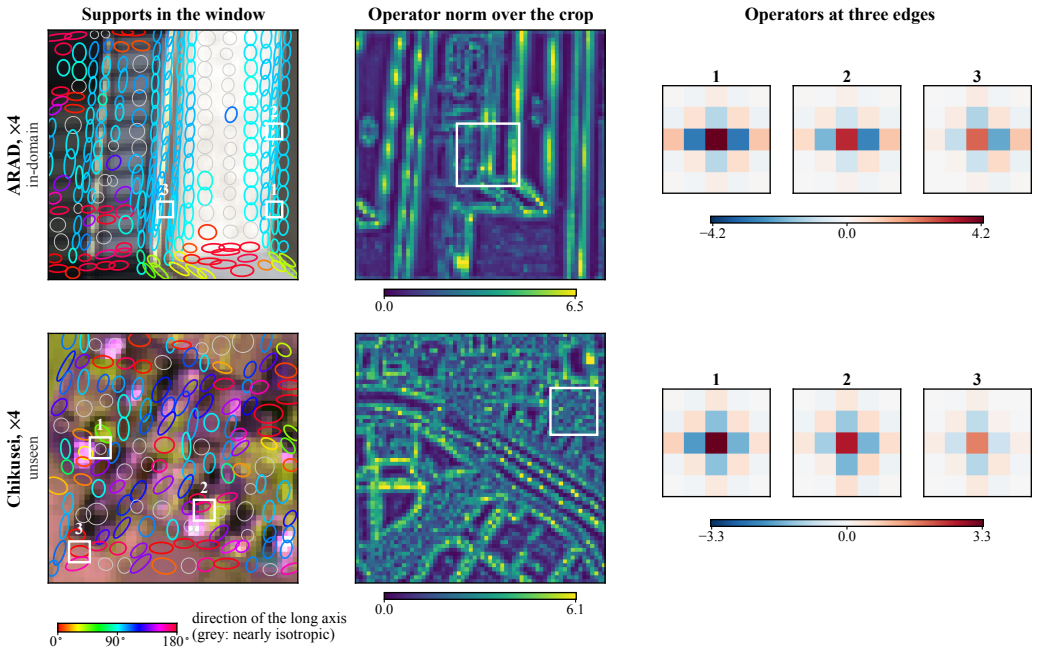


Figure 9: What the network predicts at $\times 4$. Left: the supports in the window, colored by the direction of the long axis (grey: axis ratio below 1.2). Middle: the norm of every 5×5 operator over the crop. Right: the three edge operators in the window with the largest norm.

D Inference Cost

Timing in Figure 1. Panels (a,b) profile cached multi-scale inference, whereas (d) times each output independently. Both use three $32\times32\times145$ Botswana inputs, one NVIDIA A40, batch size 1 and FP32 with TF32 disabled. For (a,b), each run requests factors $4, 6, 8, 10, 12, 14, 16$, reusing scale-independent features in both models. After two warm-up bundles, we time five bundles and select the median-total-time run per input, retaining its paired stage durations, then average across the three inputs. CSM performs feature encoding once but repeats scale conditioning and operator prediction for each output. The displayed per-output coefficients average over these seven factors and approximate this workload. Panel (d) averages per-input medians from twelve independent runs after three warm-ups at each factor.

Latency on an A40. Table 19 reports latency on one NVIDIA A40; all seven datasets, $\times 8$ and peak memory are in Table 20. On the 31-band ARAD at $\times 4$, every method takes between 9 and 33 ms and OmniHSR takes 11.8 ms; from $\times 8$ on, OmniHSR is the fastest, and at $\times 16$ it is 2.7 times faster than the fastest baseline. With more than 31 bands, the output head of a baseline is tied to 31 bands, so it runs the network 7 to 11 times over sliding windows of bands and its latency grows in proportion; OmniHSR runs once, its latency at $\times 4$ rises only from 11.8 ms at 31 bands to 22.3 ms at 145 bands, and it is 3.6 to 8.5 times faster than the strongest baseline, SRNO, and up to 36.5 times faster than the slowest. At $\times 4$, OmniHSR has the lowest peak memory of all methods. Figure 10 pushes the factor to $\times 48$: the latency of OmniHSR grows with the number of output pixels but with the lowest slope, reaching 613 ms at $\times 48$ on 128 bands, 7.3 to 32 times faster than the baselines that reach this factor; Meta-SR runs out of memory from $\times 12$ on, and SRNO and GaussianSR at $\times 48$. The time column of Table 1 and the time and memory of Table 3 are measured on one RTX 4090D with a batch size of 1 in FP32, on the first test image of each dataset, as the median of 12 synchronized runs after three warm-ups; Table 1 averages them over the twelve factors.

Table 20 is the full version of Table 19: latency and peak memory on all seven datasets at $\times 4$, $\times 8$ and $\times 16$, on the same A40 with the same timing discipline. Latency depends only on the input size

Table 19: Inference latency in ms on one NVIDIA A40 at $\times 4$ and $\times 16$ on four band counts, batch size 1, FP32, 32×32 input. **Bold**: lowest.

Method	Params (M)	ARAD (31)		Pavia U (103)		Chikusei (128)		Botswana (145)	
		$\times 4$	$\times 16$	$\times 4$	$\times 16$	$\times 4$	$\times 16$	$\times 4$	$\times 16$
LIIF	2.03	11.6	120.2	76.8	831.1	109.5	1187.1	120.5	1304.1
LTE	2.18	12.4	124.1	80.3	853.9	114.1	1224.1	127.3	1342.8
Meta-SR	6.27	23.8	271.1	171.0	1914.3	237.0	2765.6	264.6	3024.7
CiaoSR	3.11	33.1	450.4	227.0	3147.8	322.4	4489.7	354.1	4934.4
SRNO	2.48	12.3	106.0	79.0	727.9	112.1	1039.7	123.9	1142.5
GaussianSR	6.97	**8.9**	296.7	58.3	2061.8	82.1	2939.3	90.3	3238.0
OmniHSR	0.538	11.8	**39.1**	**22.1**	**148.7**	**21.4**	**129.1**	**22.3**	**135.1**
Forward passes of a baseline		1	1	7	7	10	10	11	11

and the band count, so the three 31-band datasets take the same time, and so do Pavia C and Pavia U with 102 and 103 bands.

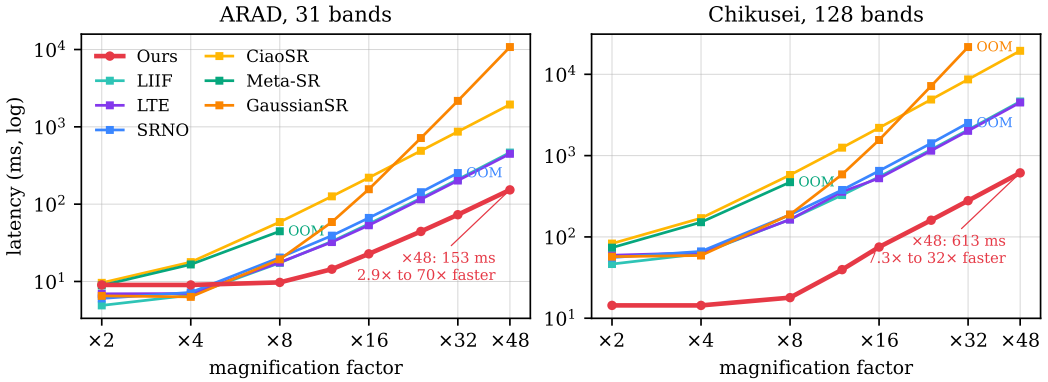


Figure 10: Latency against magnification factor on one RTX 4090D (24 GB). The same 32×32 low-resolution input is magnified by $\times 2$ to $\times 48$ (outputs of 64 to 1536 pixels), left on the 31-band ARAD and right on the 128-band Chikusei, log-log axes. The x-axis is the magnification factor of a single output and every point is one independent inference, not the accumulated time of rendering many scales from one encoding. A baseline stops where all three inputs run out of memory (OOM). The annotation gives the latency of OmniHSR at $\times 48$ and its ratio to the fastest and the slowest baseline that reach this factor.

Table 20: Full version of Table 19: latency in ms / peak allocated memory in MiB on one NVIDIA A40, batch size 1, FP32, from a 32×32 low-resolution input (median of 12 synchronized runs, averaged over three inputs), for all seven datasets at ×4, ×8 and ×16. On inputs with more than 31 bands a baseline runs its 31-band model on sliding windows of bands, which takes the number of forward passes in the last row; OmniHSR runs once. No configuration ran out of memory on the 46 GB card. **Bold**: lowest latency. Params counts the source model.

Method	Params (M)	Factor	ARAD (31)	CAVE (31)	Harvard (31)	Pavia U (103)	Pavia C (102)	Chikusei (128)	Botswana (145)
LIIF	2.030	×4	11.6 / 134	11.6 / 134	11.6 / 134	76.8 / 141	77.1 / 141	109.5 / 143	120.5 / 144
		×8	34.0 / 141	34.1 / 141	34.1 / 141	229.4 / 167	232.7 / 167	328.1 / 174	360.3 / 178
		×16	120.2 / 167	120.2 / 167	121.0 / 167	831.1 / 272	829.0 / 270	1187.1 / 296	1304.1 / 314
LTE	2.180	×4	12.4 / 123	12.4 / 123	12.5 / 123	80.3 / 130	80.9 / 130	114.1 / 132	127.3 / 133
		×8	35.2 / 130	38.9 / 130	35.5 / 130	237.1 / 156	236.6 / 156	337.3 / 163	370.6 / 167
		×16	124.1 / 156	125.5 / 156	124.2 / 156	853.9 / 261	857.9 / 259	1224.1 / 285	1342.8 / 312
Meta-SR	6.270	×4	23.8 / 2305	24.3 / 2305	24.5 / 2305	171.0 / 2313	168.7 / 2313	237.0 / 2315	264.6 / 2316
		×8	74.9 / 9122	74.6 / 9122	75.5 / 9122	553.1 / 9149	546.0 / 9149	795.9 / 9155	859.4 / 9160
		×16	271.1 / 36388	271.3 / 36388	274.3 / 36388	1914.3 / 36493	1932.1 / 36491	2765.6 / 36518	3024.7 / 36536
CiaoSR	3.110	×4	33.1 / 874	33.2 / 874	33.1 / 874	227.0 / 881	226.8 / 881	322.4 / 883	354.1 / 884
		×8	125.1 / 881	119.5 / 881	117.5 / 881	813.0 / 908	812.4 / 908	1159.7 / 914	1271.2 / 918
		×16	450.4 / 911	451.1 / 911	450.4 / 911	3147.8 / 1016	3153.5 / 1014	4489.7 / 1041	4934.4 / 1059
SRNO	2.480	×4	12.3 / 264	12.2 / 264	12.3 / 264	79.0 / 271	79.4 / 271	112.1 / 273	123.9 / 274
		×8	31.0 / 1001	31.2 / 1001	31.1 / 1001	205.3 / 1028	205.7 / 1028	292.2 / 1035	320.9 / 1039
		×16	106.0 / 3951	105.9 / 3951	105.8 / 3951	727.9 / 4056	727.4 / 4054	1039.7 / 4080	1142.5 / 4099
GaussianSR	6.970	×4	**8.9** / 273	**9.0** / 273	**9.0** / 273	58.3 / 281	58.4 / 281	82.1 / 284	90.3 / 284
		×8	33.7 / 688	34.2 / 688	33.2 / 688	231.7 / 715	227.1 / 715	328.0 / 721	358.9 / 726
		×16	296.7 / 2345	296.1 / 2345	295.8 / 2345	2061.8 / 2449	2062.5 / 2447	2939.3 / 2474	3238.0 / 2492
OmniHSR	0.538	×4	11.8 / 66	12.5 / 66	12.6 / 57	**22.1** / 110	**21.2** / 102	**21.4** / 88	**22.3** / 89
		×8	**17.6** / 209	**17.8** / 206	**16.3** / 169	**48.2** / 355	**46.7** / 347	**42.8** / 259	**45.4** / 254
		×16	**39.1** / 764	**39.2** / 761	**33.6** / 623	**148.7** / 1317	**147.8** / 1319	**129.1** / 959	**135.1** / 927
Forward passes of a baseline			1	1	1	7	7	10	11