FROM LOW-RANK TO SKETCH: CHEBYSHEV-GUIDED UNFOLDING FOR EFFICIENT PANSHARPENING

Anonymous authors
Paper under double-blind review

ABSTRACT

Pansharpening fuses a high-resolution panchromatic image with a low-resolution multispectral image. Existing methods repeatedly transform high-dimensional features, incurring cost that scales with feature dimensions and resolution. Motivated by the strong inter-band correlation and spatial-spectral redundancy of multispectral images, we design efficient sketch operators that confine the principal transformations to compact channel and spatial subspaces. To avoid the instability of repeatedly performing singular value decomposition (SVD) inside a deep network, we introduce a Chebyshev-Guided Approximation Mechanism that constructs adaptive, approximately row-orthogonal projections directly from Gram statistics. We further introduce a nullspace compensation mechanism to preserve information outside the selected subspaces. In the resulting multigrid-inspired unfolding architecture, termed OSSNet, CSS and CCS respectively provide data-adaptive spatial restriction with adjoint prolongation and compact channel routing at the coarsest level. Level-wise residual-correction blocks refine features before restriction, while coarse representations are prolonged and fused with complementary residuals during decoding. Experiments show that OSSNet achieves competitive reconstruction performance with low parameter count, measured latency, and computational cost.

1 INTRODUCTION

Pansharpening fuses a low-resolution multispectral (LRMS) image with a high-resolution panchromatic (PAN) image to produce a high-resolution multispectral (HRMS) image. Deep learning has substantially advanced this task. Despite achieving impressive reconstruction quality, existing methods typically do so at the cost of high computational complexity and large parameter counts. Recent efficient pansharpening methods fall into two categories: deployment-oriented designs using constrained operators (e.g., binary convolutions, learned look-up tables) (Hou et al., 2025; Cai et al., 2025), and compact neural designs via parameterization or distillation (Zhao et al., 2023; Howard et al., 2017; Kim et al., 2025). While effective in reducing model complexity, these approaches often sacrifice reconstruction fidelity. We instead seek to improve efficiency while retaining competitive reconstruction fidelity.

To this end, we exploit the intrinsic spatial-spectral low-rank structure of multispectral images (Wu et al., 2023; Peng et al., 2022). Sketching theory suggests that projecting high-dimensional data onto a carefully chosen low-dimensional subspace can preserve its essential structure while dramatically reducing computational cost. This motivates us to introduce bilateral sketch priors–spatial sketches that reduce spatial resolution, and channel sketches that compress channel dimensionality–to model the pansharpening problem in compact spatial-channel subspaces. Motivated by the multigrid principle that local refinement and coarse-grid correction address complementary error components, we organize the network as a learned three-level V-cycle. The resulting architecture performs most of its computation in low-dimensional spaces, leading to significant reductions in both parameters and FLOPs, while explicit complementary-residual fusion mitigates the reconstruction loss caused by dimensionality reduction.

A key challenge is that optimal sketch construction relies on SVD, which is numerically unstable when embedded in neural network training (Ionescu et al., 2015). We address this with the Chebyshev-Guided Approximation Mechanism, which replaces explicit SVD-based spectral trunca-

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.

tion with a polynomial construction that produces near-row-orthogonal projections for end-to-end learning. Since dimensionality reduction inevitably discards information, the nullspace compensation mechanism retains the complementary residual and fuses it back after sketch-space processing.

Based on this mechanism, we propose Orthogonal Subspace Sketching Network (OSSNet), a multigrid-inspired deep unfolding network for pansharpening. Experiments show that OSSNet achieves reconstruction quality competitive with state-of-the-art methods while using significantly fewer parameters and FLOPs, and exhibits stronger generalization across different satellite sensors. In summary, the main contributions of this paper are as follows:

- We formulate observation-guided feature refinement as one learned three-level multigrid-inspired V-cycle, in which adaptive spatial restriction, coarse-level processing, and prolonged complementary-residual fusion operate across resolutions.
- We introduce an SVD-free Chebyshev-Guided Approximation Mechanism that learns data-adaptive spectral filters and uses approximate row-orthogonalization to construct compact projections for end-to-end training.
- Extensive experiments demonstrate that OSSNet achieves SOTA-comparable performance with the stronger generalization capability, while using substantially fewer parameters and FLOPs.

2 Related Work

2.1 Deep Learning-Based Pansharpening

Deep learning has substantially advanced pansharpening through convolutional, attention-based, and unfolding architectures (Masi et al., 2016; Yang et al., 2017; Zhang et al., 2023; Zhou et al., 2022; Huang et al., 2025; Wang et al., 2024a; Truong et al., 2024; Wang & Ma, 2025). Despite their different designs, most repeatedly learn and transform high-dimensional features, so cost grows with feature width, spatial resolution, and the number of reconstruction stages.

Recent work has pursued efficient pansharpening along two complementary routes. The first is deployment-oriented: Hou et al. (2025) introduced BNNPan with binary wavelet convolution, while Cai et al. (2025) proposed Pan-LUT based on learned look-up tables. Such hardware-friendly operator families can provide extreme acceleration, but aggressive binarization or table-based approximation creates a tighter fidelity-efficiency trade-off, particularly when fine spatial or spectral variation must be retained. The second route redesigns conventional networks using compact parameterizations (Zhao et al., 2023), depthwise separable convolutions (Howard et al., 2017), or knowledge distillation (Kim et al., 2025). These methods improve efficiency; however, their main feature learning and transformation steps still take place in the original high-dimensional representation space.

OSSNet addresses efficiency from a different, complementary perspective. Rather than only constraining an operator or compressing a model, it exploits spatial-spectral redundancy through compact adaptive channel and local patch-wise spatial representations. The lifted sketch component is fused with its complementary residual through nullspace compensation, allowing the model to retain information outside the selected subspace.

2.2 Sketching Theory and Efficient Deep Networks

Linear sketching is supported by rigorous approximation guarantees. The Johnson-Lindenstrauss lemma shows that random projections preserve pairwise geometry with high probability, while subspace embeddings and randomized low-rank approximations retain the dominant structure needed for downstream computation (Johnson & Lindenstrauss, 1984; Woodruff, 2014; Halko et al., 2011). Learnable sketches extend this principle by adapting projection directions to the data distribution (Indyk et al., 2019; Liu et al., 2021).

The theoretical and practical benefits of sketching have motivated its use in efficient deep networks, where it commonly approximates a single expensive operator. Xiong et al. (2021) proposed Nyströmformer to approximate self-attention using the Nyström method; Chen et al. (2021) developed Skyformer by combining a Gaussian-kernel reformulation with a modified Nyström approximation; and Lee et al. (2024) introduced Proxyformer with trainable proxy tokens in place of

input-derived landmarks. These works show that compact sampled representations can substantially reduce neural computation while retaining the information required by the target operator.

In contrast, pansharpening requires repeated spatial-spectral transformations and observation-guided corrections, rather than approximation of one attention layer. OSSNet therefore uses adaptive sketches as feature-space operators: projection and its adjoint are used symmetrically, while nullspace compensation retains the complementary residual for subsequent reconstruction.

3 METHOD

This section first introduces a projected-SVD reference for constructing compact row-orthogonal sketches. We then formulate feature-domain observation consistency and organize learned residual corrections into a three-level, multigrid-inspired V-cycle comprising smoothing, spatial restriction, coarse-level processing, and prolonged correction. Finally, we instantiate the abstract operations as trainable components of OSSNet.

3.1 PRELIMINARY: PROJECTED SVD REFERENCE CONSTRUCTION

Consider a variable $\mathbf{F} \in \mathbb{R}^{C\times N}$, where C denotes the channel dimension and N the number of spatial tokens. A sketch matrix $\mathbf{Q} \in \mathbb{R}^{d\times C}$ with $d \ll C$ compresses $\mathbf{F}$ into $\mathbf{Z} = \mathbf{QF} \in \mathbb{R}^{d\times N}$. Because $\mathbf{Q}$ is real-valued, its adjoint is $\mathbf{Q}^\mathsf{T}$, and row orthogonality gives $\mathbf{QQ}^\mathsf{T} = \mathbf{I}_d$. The construction of $\mathbf{Q}$ from $\mathbf{F}$ proceeds in three steps.

Random projection. The variable $\mathbf{F}$ is first projected onto a fixed random subspace $\mathbf{\Omega} \in \mathbb{R}^{r\times C}$ with $r \ll C$:

$$\mathbf{Y} = \mathbf{\Omega F} \in \mathbb{R}^{r\times N}, \qquad \mathbf{\Omega\Omega}^\mathsf{T} = \mathbf{I}_r. \tag{1}$$

This step avoids explicit construction of the full $C \times C$ covariance matrix.

Gram matrix and SVD. The $r \times r$ Gram matrix

$$\mathbf{G} = \frac{1}{N}\mathbf{YY}^\mathsf{T} \in \mathbb{R}^{r\times r} \tag{2}$$

is computed to capture the second-order statistics of $\mathbf{F}$ in the projected subspace. A compact SVD is then applied to $\mathbf{G}$:

$$\mathbf{G} \approx \mathbf{U}_d\mathbf{\Sigma}_d\mathbf{U}_d^\mathsf{T}, \tag{3}$$

where $\mathbf{U}_d \in \mathbb{R}^{r\times d}$ contains the dominant d left singular vectors.

Sketch operator assembly. The sketch operator is assembled as

$$\mathbf{Q} = \mathbf{U}_d^\mathsf{T}\mathbf{\Omega} \in \mathbb{R}^{d\times C}, \tag{4}$$

which by construction satisfies $\mathbf{QQ}^\mathsf{T} = \mathbf{I}_d$. The sketch of $\mathbf{F}$ is given by $\mathbf{Z} = \mathbf{QF}$, and the reconstruction $\mathbf{Q}^\mathsf{T}\mathbf{Z}$ yields the optimal rank-d approximation of $\mathbf{F}$ in the projected subspace. The complementary residual $\mathbf{F} - \mathbf{Q}^\mathsf{T}\mathbf{Z}$ is explicitly retained by the nullspace compensation mechanism.

3.2 LEARNED MULTIGRID MODEL WITH ORTHOGONAL SKETCH PRIORS

Problem Formulation and Feature-Domain Lifting. Let $\mathbf{I}_{\mathrm{MS}} \in \mathbb{R}^{C\times H_{\mathrm{LR}}\times W_{\mathrm{LR}}}$ denote the low-resolution multispectral observation and $\mathbf{I}_{\mathrm{P}} \in \mathbb{R}^{1\times H\times W}$ the high-resolution panchromatic observation. Our goal is to estimate the high-resolution multispectral image $\mathbf{I}_{\mathrm{HRMS}} \in \mathbb{R}^{B\times H\times W}$. We first upsample $\mathbf{I}_{\mathrm{MS}}$ to the target spatial resolution to obtain $\mathbf{I}_{\mathrm{LMS}} \in \mathbb{R}^{C\times H\times W}$, and represent the desired image as

$$\widehat{\mathbf{I}}_{\mathrm{HRMS}} = \mathbf{I}_{\mathrm{LMS}} + \mathcal{R}(\mathbf{F}_0), \qquad \mathbf{F}_0 \in \mathbb{R}^{C\times H\times W}, \tag{5}$$

where $\mathcal{R}$ denotes the 3×3 reconstruction head that maps a feature residual to the image domain. To motivate observation-guided correction, we consider the objective

$$\mathcal{J}_0(\mathbf{F}_0) = \frac{1}{2}\left\|\mathbf{I}_{\mathrm{MS}} - \mathcal{D}_{\mathrm{MS}}(\widehat{\mathbf{I}}_{\mathrm{HRMS}})\right\|_2^2 + \frac{1}{2}\left\|\mathbf{I}_{\mathrm{P}} - \mathcal{D}_{\mathrm{P}}(\widehat{\mathbf{I}}_{\mathrm{HRMS}})\right\|_2^2 + \lambda_0\Phi_0(\mathbf{F}_0), \tag{6}$$

where $\mathcal{D}_{\mathrm{MS}}$ and $\mathcal{D}_{\mathrm{P}}$ are multispectral and panchromatic degradation operators, Φ_0 is a feature-domain prior, and $\lambda_0 > 0$ is its weight.

Learned Multigrid Hierarchy. Directly refining $\mathbf{F}_0$ at full spatial resolution is computationally expensive and can propagate slowly varying errors inefficiently. We therefore introduce a three-level hierarchy

$$\mathbf{F}_l \in \mathbb{R}^{C\times H_l\times W_l}, \qquad H_{l+1} = H_l/2, \quad W_{l+1} = W_l/2, \quad l \in \{0,1\}, \tag{7}$$

where $l = 0$ and $l = 2$ denote the finest and coarsest levels, respectively. Let $\mathcal{A}_l$ restrict a level-l feature to level $l+1$, let $\mathcal{A}_l^*$ denote its adjoint prolongation, and let $\mathbf{I}_l$ denote the identity operator on the level-l feature space. The row-orthogonal construction gives

$$\mathcal{A}_l : \mathbb{R}^{C\times H_l\times W_l} \to \mathbb{R}^{C\times H_{l+1}\times W_{l+1}}, \qquad \mathcal{A}_l\mathcal{A}_l^* = \mathbf{I}_{l+1}, \quad l \in \{0,1\}. \tag{8}$$

Consequently, $\mathcal{A}_l^*\mathcal{A}_l$ projects a fine-level feature onto the subspace represented at level $l+1$, whereas $(\mathbf{I}_l - \mathcal{A}_l^*\mathcal{A}_l)\mathbf{F}_l$ is its complementary fine-scale component.

Level-Wise Residual Smoothing. At a fixed level l, we suppress the level indices on the observations and degradation maps. Let k index the local GCM update, so that $\mathbf{F}_l^k$ and $\mathbf{F}_l^{k+1}$ denote its input and output, respectively. Let $\mathcal{D}_{\mathrm{MS}}$ and $\mathcal{D}_{\mathrm{P}}$ denote the corresponding degradation operators, and let $\mathcal{R}$ denote the reconstruction head. The reconstruction head maps $\mathbf{F}_l^k$ to the current image-domain estimate $\mathbf{Y}^k$. The resulting observation residuals are

$$\mathbf{Y}^k = \mathcal{R}(\mathbf{F}_l^k), \qquad \mathbf{r}_{\mathrm{MS}}^k = \mathbf{I}_{\mathrm{MS}} - \mathcal{D}_{\mathrm{MS}}(\mathbf{Y}^k), \qquad \mathbf{r}_{\mathrm{P}}^k = \mathbf{I}_{\mathrm{P}} - \mathcal{D}_{\mathrm{P}}(\mathbf{Y}^k). \tag{9}$$

Here, $\mathcal{R}^*$ and $\mathcal{D}_{\mathrm{MS}}^*$ are the associated adjoints, whereas $\widetilde{\mathcal{D}}_{\mathrm{P}}^*$ is a learned PAN back-projection. The learned map $\mathcal{G}_l$ fuses $\mathbf{g}_{\mathrm{MS}}^k$ and $\mathbf{g}_{\mathrm{P}}^k$ with $\mathbf{F}_l^k$ to predict $\Delta\mathbf{F}_l^k$. The scalar τ_l weights this correction, and the updated feature is $\mathbf{F}_l^{k+1}$. The level-wise smoothing is therefore:

$$\begin{aligned} \mathbf{g}_{\mathrm{MS}}^k &= \mathcal{R}^*\mathcal{D}_{\mathrm{MS}}^*(\mathbf{r}_{\mathrm{MS}}^k), \qquad \mathbf{g}_{\mathrm{P}}^k = \mathcal{R}^*\widetilde{\mathcal{D}}_{\mathrm{P}}^*(\mathbf{r}_{\mathrm{P}}^k), \\ \Delta\mathbf{F}_l^k &= \mathcal{G}_l(\mathbf{g}_{\mathrm{MS}}^k, \mathbf{g}_{\mathrm{P}}^k, \mathbf{F}_l^k), \qquad \mathbf{F}_l^{k+1} = \mathbf{F}_l^k + \tau_l\Delta\mathbf{F}_l^k. \end{aligned} \tag{10}$$

The pre-smoother subsequently refines $\mathbf{F}_l^{k+1}$, and we retain the same notation for its output.

Restriction and Coarse-Level Processing. After pre-smoothing, the feature estimate is transferred to the next coarser level:

$$\mathbf{F}_{l+1}^k = \mathcal{A}_l\mathbf{F}_l^{k+1}, \qquad l \in \{0,1\}. \tag{11}$$

At the coarsest spatial level, we further separate the feature into a compact channel subspace and its complementary residual. Let $\mathbf{Q}_{\mathrm{c}} \in \mathbb{R}^{d\times C}$ be a row-orthogonal channel projection satisfying $\mathbf{Q}_{\mathrm{c}}\mathbf{Q}_{\mathrm{c}}^{\mathsf{T}} = \mathbf{I}_d$, where $d \ll C$. Defining $\mathbf{Z}_{\mathrm{c}}$ as the compact representation and $\mathbf{N}_{\mathrm{c}}$ as the complementary residual gives

$$\mathbf{Z}_{\mathrm{c}} = \mathbf{Q}_{\mathrm{c}}\mathbf{F}_2^{k+1}, \qquad \mathbf{N}_{\mathrm{c}} = (\mathbf{I}_C - \mathbf{Q}_{\mathrm{c}}^{\mathsf{T}}\mathbf{Q}_{\mathrm{c}})\mathbf{F}_2^{k+1}, \qquad \mathbf{F}_2^{k+1} = \mathbf{Q}_{\mathrm{c}}^{\mathsf{T}}\mathbf{Z}_{\mathrm{c}} + \mathbf{N}_{\mathrm{c}}. \tag{12}$$

Let $\mathcal{B}$ process the compact representation. The lifted compact response and the complementary residual are then combined by a learned fusion operator, denoted by Fuse_2, to produce the coarsest decoded feature $\mathbf{U}_2$:

$$\mathbf{U}_2 = \mathrm{Fuse}_2\big(\mathbf{Q}_{\mathrm{c}}^{\mathsf{T}}\mathcal{B}(\mathbf{Z}_{\mathrm{c}}), \mathbf{N}_{\mathrm{c}}\big)\,. \tag{13}$$

Prolongation and Coarse-Grid Correction. Starting from $\mathbf{U}_2$, the hierarchy is traversed in reverse to obtain the decoded features $\mathbf{U}_1$ and $\mathbf{U}_0$. Denote the spatial complementary residual by $\mathbf{N}_l = (\mathbf{I}_l - \mathcal{A}_l^*\mathcal{A}_l)\mathbf{F}_l^{k+1}$. Under ideal additive fusion, nullspace completion is equivalent to additive coarse-grid correction:

$$\mathbf{U}_l = \mathcal{A}_l^*\mathbf{U}_{l+1} + \mathbf{N}_l = \mathbf{F}_l^{k+1} + \mathcal{A}_l^*\big(\mathbf{U}_{l+1} - \mathcal{A}_l\mathbf{F}_l^{k+1}\big)\,, \qquad l = 1, 0. \tag{14}$$

In the learned realization, Fuse_l combines each prolonged coarse feature with its complementary residual to form the decoded feature:

$$\mathbf{U}_l = \mathrm{Fuse}_l(\mathcal{A}_l^*\mathbf{U}_{l+1}, \mathbf{N}_l)\,, \qquad l = 1, 0. \tag{15}$$

A post-smoother subsequently refines $\mathbf{U}_l$, and we retain the same notation for its output. Equations (10)–(15) define one learned three-level V-cycle. The formulation follows the restriction–coarse processing–prolongation pattern of nonlinear multigrid while leaving the residual correction, coarse-level processing, and fusion maps learnable.

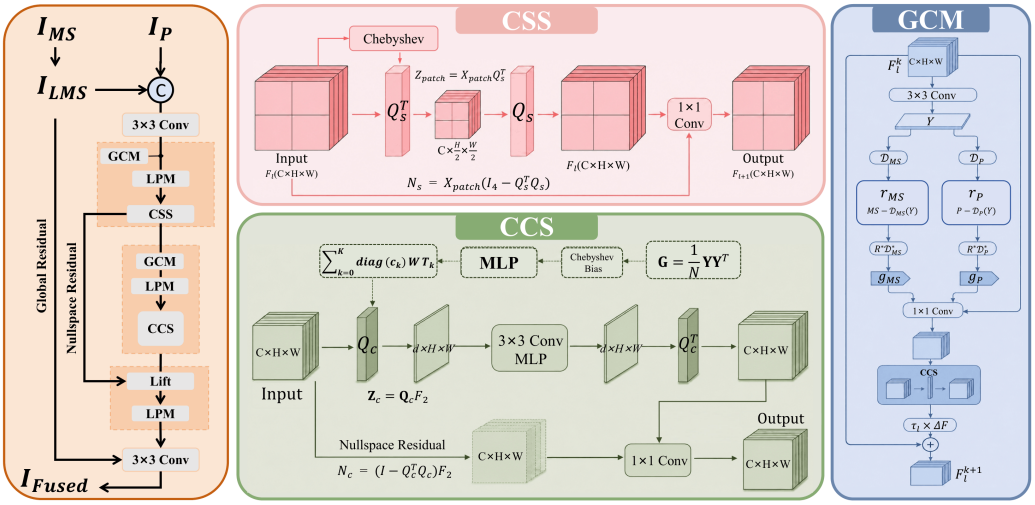


Figure 1: Overview of OSSNet. CSS performs two adaptive spatial restrictions and their adjoint lifts, while spatial complementary residuals are fused in the decoder. CCS performs bottleneck channel sketching with complementary-residual fusion. At each scale, GCM back-projects the MS and PAN residuals into $\mathbf{g}_{\mathrm{MS}}^{k}$ and $\mathbf{g}_{\mathrm{P}}^{k}$ and updates $\mathbf{F}_{l}^{k}$ to $\mathbf{F}_{l}^{k+1}$.

3.3 Network Architecture Design

The preceding formulation specifies the computational roles of smoothing, restriction, coarse-level processing, prolongation, and complementary-residual fusion. Our proposed network, **OSSNet**, instantiates each of these components as a learnable neural module, organizing the learnable modules as one three-level multigrid V-cycle.

Overall Pipeline. As illustrated in Figure 1, the network executes one learned V-cycle over three spatial levels. The encoder downsamples via the Chebyshev Spatial Sketch (CSS), which implements orthogonal spatial restriction. This reduces the spatial resolution at deeper stages, allowing subsequent modules to operate on lower-resolution feature maps for computational efficiency. At each scale, a Gradient Compensation Module (GCM) updates $\mathbf{F}_{l}^{k}$ to $\mathbf{F}_{l}^{k+1}$, followed by a Lightweight Prior Module (LPM). At the coarsest scale, the primary Chebyshev Channel Sketch (CCS) compresses the channel dimension into a low-dimensional sketch; its compact representation is refined by a lightweight processor and then fused with the channel complementary residual to produce $\mathbf{U}_2$. The decoder reverses the path via the adjoint CSS, fuses the spatial complementary residuals at each stage, and applies LPMs to produce $\mathbf{U}_1$ and $\mathbf{U}_0$. The decoded finest-level state $\mathbf{U}_0$ realizes the latent solution in Eq. (5), yielding $\widehat{\mathbf{I}}_{\mathrm{HRMS}} = \mathbf{I}_{\mathrm{LMS}} + \mathcal{R}(\mathbf{U}_0)$. Together, CSS and CCS reduce the effective dimensionality of both spatial and channel domains throughout the network, enabling efficient optimization on compact representations while preserving complementary residuals through nullspace compensation.

Chebyshev-Guided Approximation Mechanism. To realize the bilateral sketch priors introduced in the preliminary construction above, we design two complementary operators: the Chebyshev Channel Sketch (CCS) compresses the channel dimension, while the Chebyshev Spatial Sketch (CSS) implements dynamic restriction and prolongation for spatial scale transitions. Both operators use the Chebyshev-Guided Approximation Mechanism.

In classical sketching, the optimal projection would require a compact SVD of $\mathbf{G}$ to retain the d dominant eigenvectors, as expressed in Eq. (3). This step is inherently a spectral truncation preserving directions associated with the largest eigenvalues while discarding the remaining directions. Although algebraically optimal, gradients of singular vectors can become ill-conditioned when singular values are repeated or closely clustered, which may destabilize end-to-end training (Ionescu et al., 2015). We therefore seek a learnable surrogate for this spectral selection step without explicitly differentiating through SVD.

Polynomial matrix functions provide a principled route. For a symmetric matrix $\mathbf{G} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\mathsf{T}$, a polynomial matrix function satisfies $p(\mathbf{G}) = \mathbf{V}p(\mathbf{\Lambda})\mathbf{V}^\mathsf{T}$, thus preserving the eigenvectors while mapping every eigenvalue λ to $p(\lambda)$ (Higham, 2008). Chebyshev polynomials provide an efficient basis for constructing such spectral responses without eigendecomposition (Defferrard et al., 2016). Building on these insights, we design the Chebyshev-Guided Approximation Mechanism and replace the hard top-d cutoff with a smooth, data-adaptive polynomial response.

Concretely, given a feature map $\mathbf{F} \in \mathbb{R}^{C\times H\times W}$, we flatten it into $\widetilde{\mathbf{F}} \in \mathbb{R}^{C\times N}$ with $N = HW$. Let $\mathbf{1}_N \in \mathbb{R}^N$ be the all-ones vector, $\boldsymbol{\mu} = N^{-1}\widetilde{\mathbf{F}}\mathbf{1}_N \in \mathbb{R}^C$ the spatial mean, and $\widetilde{\mathbf{F}}_\mathrm{c} = \widetilde{\mathbf{F}} - \boldsymbol{\mu}\mathbf{1}_N^\mathsf{T}$ the centered feature matrix. The channel covariance is

$$\mathbf{G} = \frac{1}{N}\widetilde{\mathbf{F}}_\mathrm{c}\widetilde{\mathbf{F}}_\mathrm{c}^\mathsf{T} \in \mathbb{R}^{C\times C}. \tag{16}$$

Let K denote the polynomial order. To distinguish polynomial degree from the GCM update index k, we index the Chebyshev basis by j. The basis matrices $\{\mathbf{T}_j\}_{j=0}^K$ are generated by

$$\mathbf{T}_0 = \mathbf{I}_C, \qquad \mathbf{T}_1 = \mathbf{G}, \qquad \mathbf{T}_j = 2\mathbf{G}\mathbf{T}_{j-1} - \mathbf{T}_{j-2} \quad (2 \le j \le K). \tag{17}$$

To construct a d-dimensional sketch, we introduce row-orthogonal anchor directions $\mathbf{W} \in \mathbb{R}^{d\times C}$ satisfying $\mathbf{W}\mathbf{W}^\mathsf{T} = \mathbf{I}_d$. A coefficient predictor takes the trace statistics $[\log\mathrm{tr}(\mathbf{G}), \mathrm{tr}(\mathbf{T}_1), \ldots, \mathrm{tr}(\mathbf{T}_K)]$ as input and returns $\mathbf{\Gamma} = [\boldsymbol{\gamma}_0, \ldots, \boldsymbol{\gamma}_K] \in \mathbb{R}^{d\times(K+1)}$, where $\boldsymbol{\gamma}_j \in \mathbb{R}^d$ modulates the response to the j-th polynomial component. The raw sketch matrix is

$$\mathbf{Q}^\mathrm{raw} = \sum_{j=0}^{K} \mathrm{diag}(\boldsymbol{\gamma}_j)\mathbf{W}\mathbf{T}_j. \tag{18}$$

The matrix $\mathrm{diag}(\boldsymbol{\gamma}_j)$ independently modulates the d anchor responses. The predicted matrix is row-orthogonalized by a few Newton–Schulz iterations to obtain $\mathbf{Q}$ satisfying $\mathbf{Q}\mathbf{Q}^\mathsf{T} \approx \mathbf{I}_d$. The resulting $\mathbf{Q}$ and $\mathbf{Q}^\mathsf{T}$ serve as the analysis and synthesis operators, respectively.

Chebyshev Spatial Sketch (CSS) Implementation. The CSS instantiates $\mathcal{A}_l$ and its adjoint $\mathcal{A}_l^*$ by partitioning each feature map into non-overlapping 2×2 patches and predicting a projection direction for every patch. Let $\mathbf{X}_\mathrm{patch} \in \mathbb{R}^{C\times4}$ denote one patch and let $\mathbf{\Omega}_\mathrm{s} \in \mathbb{R}^{C'\times C}$ be the fixed random channel projection used only to estimate its 4×4 spatial covariance. The resulting spatial sketch $\mathbf{Q}_\mathrm{s} \in \mathbb{R}^{1\times4}$ satisfies $\mathbf{Q}_\mathrm{s}\mathbf{Q}_\mathrm{s}^\mathsf{T} = 1$. Restriction gives $\mathbf{Z}_\mathrm{patch} = \mathbf{X}_\mathrm{patch}\mathbf{Q}_\mathrm{s}^\mathsf{T} \in \mathbb{R}^{C\times1}$, prolongation uses $\mathbf{Q}_\mathrm{s}$, and the complementary residual is $\mathbf{X}_\mathrm{patch}(\mathbf{I}_4 - \mathbf{Q}_\mathrm{s}^\mathsf{T}\mathbf{Q}_\mathrm{s})$.

Chebyshev Channel Sketch (CCS) Implementation. The CCS instantiates the channel analysis–synthesis pair $\mathbf{Q}_\mathrm{c}/\mathbf{Q}_\mathrm{c}^\mathsf{T}$ introduced in Eq. (12). It aggregates second-order statistics over all spatial tokens and produces one projection $\mathbf{Q}_\mathrm{c} \in \mathbb{R}^{d\times C}$, with $d \ll C$, for each input feature map $\mathbf{F}$. The compact representation $\mathbf{Z}_\mathrm{c} = \mathbf{Q}_\mathrm{c}\mathbf{F}$ is refined by a 3×3 convolution and a channel MLP, lifted by $\mathbf{Q}_\mathrm{c}^\mathsf{T}$, and fused with the complementary residual $\mathbf{N}_\mathrm{c} = (\mathbf{I}_C - \mathbf{Q}_\mathrm{c}^\mathsf{T}\mathbf{Q}_\mathrm{c})\mathbf{F}$ through a 1×1 convolution. Thus, CCS routes reconstruction-relevant components through a compact branch while retaining the complementary residual through nullspace compensation. It is used both at the coarsest level and within each residual-correction map, with the latter using a smaller sketch dimension for efficiency.

3.3.1 Learned Smoothing: Gradient Compensation and Lightweight Prior Refinement

The Gradient Compensation Module (GCM) instantiates $\mathcal{G}_l$ in Eq. (10). The reconstruction head $\mathcal{R}$ is implemented by a 3×3 convolution, while $\mathcal{D}_\mathrm{MS}$ and $\mathcal{D}_\mathrm{P}$ are learnable degradation maps at the current level. The MS branch applies $\mathcal{D}_\mathrm{MS}^*$ and $\mathcal{R}^*$ to produce $\mathbf{g}_\mathrm{MS}$. The PAN branch applies the learned back-projection $\widetilde{\mathcal{D}}_\mathrm{P}^*$ and $\mathcal{R}^*$ to produce $\mathbf{g}_\mathrm{P}$. A 1×1 fusion combines these signals with $\mathbf{F}_l^k$, after which an auxiliary channel sketch produces $\Delta\mathbf{F}_l^k$ and updates the feature to $\mathbf{F}_l^{k+1}$. The Lightweight Prior Module (LPM) implements the pre- and post-smoothing operations. It comprises GroupNorm, a grouped 3×3 convolution, GELU activation (Hendrycks & Gimpel, 2016), a 1×1 convolution, and a learnable residual scale.

Table 1: Average metrics on reduced-resolution WV3, QB, and GF2 samples, and zero-shot cross-sensor transfer on WV2. Best results in **bold**, second-best underlined.

Method	WV3 Reduce			QB Reduce			GF2 Reduce			WV2 Zero-Shot			Params (M)↓	FLOPs (G)↓
	PSNR↑	ERGAS↓	SAM↓	PSNR↑	ERGAS↓	SAM↓	PSNR↑	ERGAS↓	SAM↓	PSNR↑	ERGAS↓	SAM↓		
TDNet	36.03	2.54	3.64	37.17	4.15	4.58	47.27	1.02	1.23	26.68	9.23	6.04	0.49	20.00
DISPNet	36.80	2.33	3.42	36.83	4.33	4.47	48.76	0.86	1.02	28.46	4.61	5.81	0.33	25.60
UTeRM	36.23	2.48	3.49	37.49	3.98	4.88	47.84	0.96	1.13	28.02	4.83	5.90	4.39	301.67
DCINN	**37.22**	**2.24**	3.30	38.42	3.57	4.22	49.28	0.81	0.98	27.93	4.94	5.82	0.49	13.62
WFA-Net	37.03	2.28	3.35	38.26	3.63	4.27	48.56	0.88	1.05	28.08	4.87	6.06	0.53	14.33
ARConv	36.58	2.38	3.41	37.65	3.90	4.58	45.72	1.19	1.36	28.61	4.51	5.58	4.79	61.57
BNNPan	36.00	2.54	3.63	37.66	3.90	5.07	40.77	1.09	1.26	27.43	5.16	6.47	0.25	15.28
DAPNet	37.12	2.26	3.31	37.42	4.02	4.37	**49.44**	**0.80**	**0.95**	29.00	4.33	5.69	2.33	135.64
OSSNet	**37.22**	**2.23**	**3.29**	**38.67**	**3.47**	**4.16**	49.34	**0.80**	**0.95**	**30.24**	**3.68**	**5.00**	**0.17**	**4.10**

Remark. Two key differences distinguish OSSNet from U-Net (Ronneberger et al., 2015). First, U-Net copies full encoder features to the decoder, whereas OSSNet transmits the complementary residuals $(\mathbf{I}_l - \mathcal{A}_l^* \mathcal{A}_l)\mathbf{F}_l^{k+1}$ through nullspace compensation. Second, U-Net uses fixed pooling and interpolation, whereas OSSNet uses adaptive restriction $\mathcal{A}_l$ and its adjoint prolongation $\mathcal{A}_l^*$.

3.4 Loss Function and Training Objective

OSSNet is trained end-to-end with an ℓ_1 reconstruction loss and an orthogonality regularizer:

$$\mathcal{L} = \left\| \widehat{\mathbf{I}}_{\mathrm{HRMS}} - \mathbf{I}_{\mathrm{HRMS}} \right\|_1 + 10^{-4} \mathcal{L}_{\mathrm{orth}}, \tag{19}$$

where $\mathcal{C}_{\mathrm{sk}}$ is the index set of the four channel-sketch instances and

$$\mathcal{L}_{\mathrm{orth}} = \frac{1}{|\mathcal{C}_{\mathrm{sk}}|} \sum_{m \in \mathcal{C}_{\mathrm{sk}}} \frac{1}{d_m^2} \left\| \mathbf{Q}_m \mathbf{Q}_m^{\mathsf{T}} - \mathbf{I}_{d_m} \right\|_1, \tag{20}$$

with d_m denoting the row dimension of $\mathbf{Q}_m$.

4 Experiments

4.1 Datasets and Metrics

We evaluate the model on three widely used satellite datasets: WorldView-3 (WV3), Gaofen-2 (GF2), and QuickBird (QB). For each dataset, we follow the Wald protocol (Wald, 2000) to construct reduced-resolution training and test sets with reference images. We also evaluate zero-shot transfer to WorldView-2 (WV2), using WV3-trained models without fine-tuning. Reduced-resolution and zero-shot experiments report Peak Signal-to-Noise Ratio (PSNR), ERGAS (Wald, 2000), and Spectral Angle Mapper (SAM) (Yuhas et al., 1992; Kruse et al., 1993). Full-resolution evaluation reports HQNR and its spectral and spatial distortion terms, D_λ and D_s (Alparone et al., 2008).

4.2 Comparison with Representative Methods

We compare OSSNet with eight representative pansharpening methods: TDNet (Zhang et al., 2023), DISPNet (Wang et al., 2024a), UTeRM (Truong et al., 2024), DCINN (Wang et al., 2024b), WFA-Net (Huang et al., 2025), ARConv (Wang et al., 2025), BNNPan (Hou et al., 2025), and DAP-Net (Wang & Ma, 2025).

Quantitative Evaluation. As reported in Tables 1 and 2, OSSNet achieves the best reduced-resolution results on all QB and WV3 metrics. On GF2, it ranks second in PSNR and ties the best ERGAS and SAM. In zero-shot transfer from WV3 to WV2, OSSNet leads all three metrics, demonstrating strong cross-sensor generalization. Under the unified full-resolution evaluation, OSSNet obtains HQNR values of 0.952, 0.905, and 0.893 on WV3, QB, and GF2, respectively. It ties for second in WV3 HQNR and for the lowest WV3 D_λ. It ranks second in QB HQNR and fourth in GF2 HQNR. With only 0.17M parameters and 4.10G FLOPs, OSSNet nevertheless achieves the lowest inference latency of 4.96 ms, retaining a favorable accuracy–efficiency trade-off.

Table 2: Full-resolution evaluation on WV3, QB, and GF2. Completed, protocol-consistent reruns replace the corresponding reported values and use the same HQNR implementation. Best results are **bolded**, second-best underlined. OOM = out of memory.

Method	WV3 Full			QB Full			GF2 Full			Latency (ms)↓	Peak Mem. (MB)↓
	HQNR↑	D_λ↓	D_s↓	HQNR↑	D_λ↓	D_s↓	HQNR↑	D_λ↓	D_s↓		
TDNet	0.925	0.032	0.045	0.880	0.080	0.044	0.879	0.055	0.071	6.90	84.20
DISPNet	0.947	0.027	0.027	0.881	0.081	0.043	0.885	0.048	0.071	29.70	101.90
UTeRM	0.935	0.035	0.033	0.894	**0.029**	0.081	0.823	0.105	0.083	214.37	220.18
DCINN	0.952	**0.023**	0.026	0.890	0.043	0.070	0.887	0.038	0.077	10.27	**56.15**
WFA-Net	0.950	0.027	**0.024**	0.855	0.054	0.096	**0.902**	0.041	**0.059**	OOM	OOM
ARConv	**0.953**	**0.023**	0.025	**0.915**	0.052	0.035	0.897	**0.034**	0.071	92.56	671.58
BNNPan	0.928	0.036	0.037	0.834	0.095	0.078	0.890	0.043	0.070	31.60	159.18
DAPNet	0.947	0.026	0.028	0.895	0.082	**0.026**	0.894	0.045	0.065	76.52	183.85
OSSNet	0.952	**0.023**	0.027	0.905	0.045	0.054	0.893	0.039	0.070	**4.96**	95.08

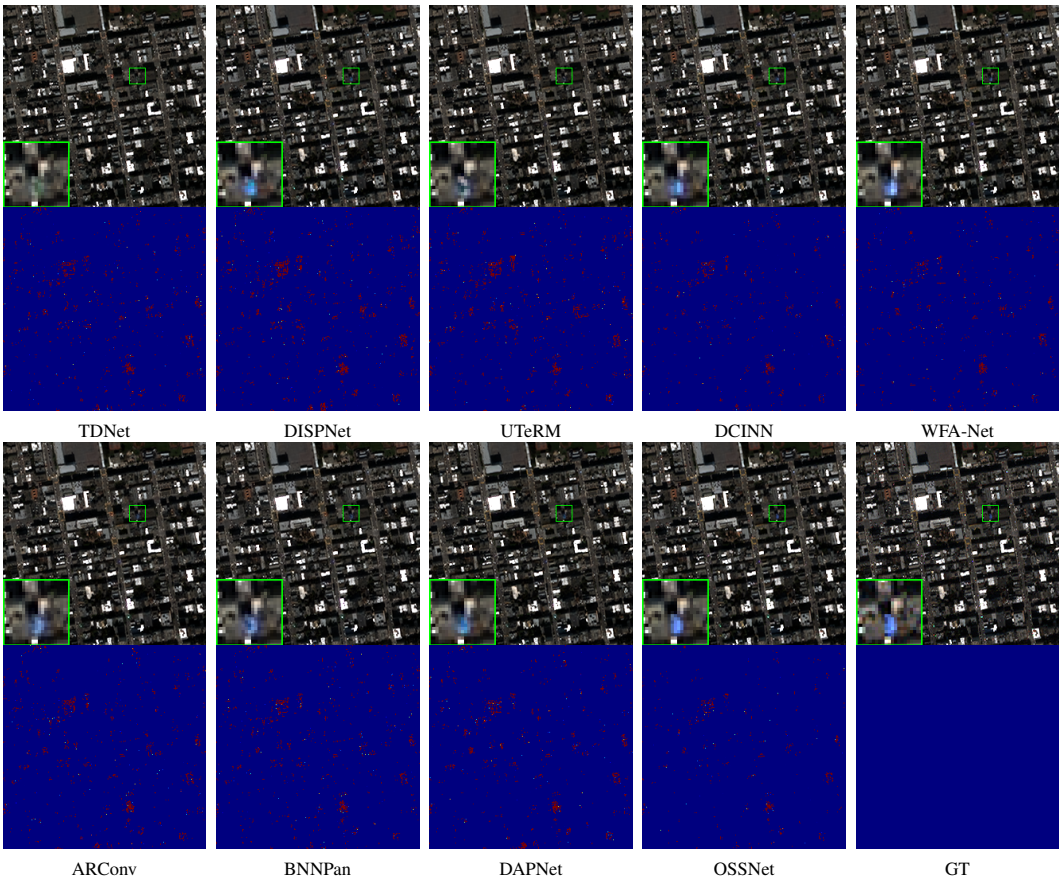


Figure 2: Qualitative comparison on a representative reduced-resolution sample. Methods are arranged in two rows of five; each entry shows the reconstructed image above its absolute error map.

Qualitative Visualization. Visual comparisons further substantiate these findings. Figure 2 shows that OSSNet retains clear structural boundaries while producing visually low residual intensity in the reduced-resolution example. In the full-resolution example of Fig. 3, OSSNet exhibits fewer visible color and saturation artifacts in vegetation regions. All reconstructed images and error maps are displayed using identical intensity ranges to ensure a fair visual comparison.



Figure 3: Qualitative comparison on a representative full-resolution sample.

Table 3: Ablation results on reduced-Resolution WV3 and zero-shot WV2.

Variant	PSNR (dB)↑		Params (M)↓	FLOPs (G)↓
	WV3	WV2		
w/o GCM	36.89	29.29	**0.09**	**2.73**
w/o Nullspace Comp.	**37.22**	29.77	0.17	3.92
w/o CSS	37.19	28.77	0.19	4.83
w/o CCS	36.89	27.92	0.14	4.26
w/o Chebyshev	37.04	29.34	0.13	4.08
Full OSSNet	**37.22**	**30.24**	0.17	4.10

Table 4: Sensitivity to the orthogonality-loss weight (original ablation protocol).

λ_{orth}	WV3 PSNR↑	WV2 PSNR↑	WV3 HQNR↑
0	37.17	29.61	0.961
10^{-5}	37.18	29.93	0.963
10^{-4}	**37.22**	30.24	**0.970**
10^{-3}	37.16	29.93	0.960
10^{-2}	37.15	**30.34**	0.964
10^{-1}	37.12	29.54	0.961

4.3 Ablation Studies

We evaluate five controlled variants on reduced-resolution WV3 and zero-shot WV2 (Table 3). The w/o GCM variant removes all GCMs, while w/o Nullspace Comp. replaces complementary-residual paths with standard residual connections. The w/o CSS and w/o CCS variants replace the corresponding sketches with PixelShuffle-based spatial mappings and separate 1×1 channel mappings, respectively. The w/o Chebyshev variant instead constructs projections by repeated SVD with scale normalization, diagonal perturbation ($\epsilon = 10^{-5}$), and eigen-gap clamping.

Removing GCM reduces complexity to 0.09M parameters and 2.73G FLOPs, but decreases WV3/WV2 PSNR by 0.33/0.95 dB. Removing nullspace compensation leaves WV3 unchanged but costs 0.47 dB on WV2. Removing CSS increases both parameters and FLOPs while reducing WV2 PSNR by 1.47 dB. Removing CCS costs 2.32 dB on WV2 and raises computation to 4.26G FLOPs despite using fewer parameters. Repeated SVD reduces WV3/WV2 PSNR by 0.18/0.90 dB. These results support observation-guided correction, compact bilateral sketching, and complementary-residual preservation, particularly under sensor shift.

4.4 Validation of Orthogonality Constraints

To isolate the effect of orthogonality regularization, we varied only λ_{orth} while holding all other settings fixed. Table 4 reports WV3 PSNR and HQNR, together with zero-shot WV2 PSNR, across different values of λ_{orth}. At $\lambda_{\text{orth}} = 10^{-4}$, OSSNet achieved the highest WV3 PSNR (37.22 dB) and HQNR (0.970), while yielding a zero-shot WV2 PSNR of 30.24 dB. In contrast, $\lambda_{\text{orth}} = 10^{-2}$ achieved the highest WV2 PSNR (30.34 dB), but reduced WV3 PSNR and HQNR to 37.16 dB and 0.964, respectively. This trade-off indicates that moderate orthogonality regularization improves the cross-sensor robustness. We select $\lambda_{\text{orth}} = 10^{-4}$ in all remaining experiments.

5 Conclusion

We propose OSSNet, a multigrid-inspired deep unfolding network that organizes bilateral sketch priors into one learned three-level V-cycle. Its Chebyshev-Guided Approximation Mechanism constructs data-adaptive, approximately row-orthogonal spatial and channel projections from Gram statistics without repeated SVD. These projections concentrate computation in compact subspaces, while nullspace compensation preserves complementary residuals and multiscale GCMs enforce observation consistency. Across standard benchmarks, OSSNet achieves competitive reconstruction quality with the fewest parameters, lowest FLOPs and inference latency, and strong cross-sensor generalization.

AI USE STATEMENT

We used AI to assist with manuscript drafting, refinement of the conceptual explanation and mathematical descriptions of the framework, feedback on ablation design and experimental protocols. We also used it to check related literature and manuscript structure, improve readability, format references and tables, and edit result-summary code. The authors retain responsibility for reviewing and verifying all AI-assisted content and for the final text, claims, references, code, and numerical reports.

REFERENCES

Luciano Alparone, Bruno Aiazzi, Stefano Baronti, Andrea Garzelli, Filippo Nencini, and Massimo Selva. Multispectral and panchromatic data fusion assessment without reference. *Photogrammetric Engineering and Remote Sensing*, 74(2):193–200, 2008. doi: 10.14358/PERS.74.2.193.

Zhongnan Cai, Yingying Wang, Hui Zheng, Panwang Pan, Zixu Lin, Ge Meng, Chenxin Li, Chunming He, Jiaxin Xie, Yunlong Lin, Junbin Lu, Yue Huang, and Xinghao Ding. Pan-LUT: Efficient pan-sharpening via learnable look-up tables. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://openreview.net/forum?id=OzdAnGHEPx.

Yifan Chen, Qi Zeng, Heng Ji, and Yun Yang. Skyformer: Remodel self-attention with Gaussian kernel and Nyström method. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

M. Defferrard, X. Bresson, and P. Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29, pp. 3844–3852, 2016.

Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53 (2):217–288, 2011. doi: 10.1137/090771806.

D. Hendrycks and K. Gimpel. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.

Nicholas J. Higham. *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008. doi: 10.1137/1.9780898717778.

Junming Hou, Xiaoyu Chen, Ran Ran, Xiaofeng Cong, Xinyang Liu, Jian Wei You, and Liang-Jian Deng. Binarized neural network for multi-spectral image fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2236–2245, 2025. doi: 10.1109/CVPR52734.2025.00214.

Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. URL https://arxiv.org/abs/1704.04861.

Jie Huang, Rui Huang, Jinghao Xu, Siran Peng, Yule Duan, and Liang-Jian Deng. Wavelet-assisted multi-frequency attention network for pansharpening. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 3662–3670, 2025. doi: 10.1609/aaai.v39i4.32381.

Piotr Indyk, Ali Vakilian, and Yang Yuan. Learning-based low-rank approximations. In *Advances in Neural Information Processing Systems*, volume 32, pp. 7400–7410, 2019.

Catalin Ionescu, Orestis Vantzos, and Cristian Sminchisescu. Matrix backpropagation for deep networks with structured layers. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2965–2973, 2015. doi: 10.1109/ICCV.2015.339.

William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In *Contemporary Mathematics*, volume 26, pp. 189–206. American Mathematical Society, 1984. doi: 10.1090/conm/026/737400.

Sungpyo Kim, Jeonghyeok Do, Jaehyup Lee, and Munchurl Kim. U-Know-DiffPAN: An uncertainty-aware knowledge distillation diffusion framework with details enhancement for PAN-sharpening. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23069–23079, 2025. doi: 10.1109/CVPR52734.2025.02148.

Fred A. Kruse, Alexander B. Lefkoff, Joseph W. Boardman, Karl B. Heidebrecht, A. T. Shapiro, P. J. Barloon, and Alexander F. H. Goetz. The spectral image processing system (sips): Interactive visualization and analysis of imaging spectrometer data. *Remote Sensing of Environment*, 44 (2–3):145–163, 1993. doi: 10.1016/0034-4257(93)90013-N.

Sangho Lee, Hayun Lee, and Dongkun Shin. Proxyformer: Nyström-based linear transformer with trainable proxy tokens. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 13418–13426, 2024. doi: 10.1609/aaai.v38i12.29244.

Simin Liu, Tianrui Liu, Ali Vakilian, Yulin Wan, and David P. Woodruff. A framework for learned CountSketch. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=RDiiCiIH3_B.

Giuseppe Masi, Davide Cozzolino, Luisa Verdoliva, and Giuseppe Scarpa. Pansharpening by convolutional neural networks. *Remote Sensing*, 8(7):594, 2016. doi: 10.3390/rs8070594.

Jiangtao Peng, Weiwei Sun, Heng-Chao Li, Wei Li, Xiangchao Meng, Chiru Ge, and Qian Du. Low-rank and sparse representation for hyperspectral image processing: A review. *IEEE Geoscience and Remote Sensing Magazine*, 10(1):10–43, 2022. doi: 10.1109/MGRS.2021.3075491.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 234–241. Springer, 2015.

Thanh Nhat Mai Truong, Edmund Y. Lam, and Chul Lee. Deep unfolding tensor rank minimization with generalized detail injection for pansharpening. *IEEE Transactions on Geoscience and Remote Sensing*, 62, 2024. doi: 10.1109/TGRS.2024.3392215.

Lucien Wald. Quality of high resolution synthesised images: Is there a simple criterion? In *Proceedings of the Third Conference Fusion of Earth Data*, pp. 99–103, Sophia Antipolis, France, 2000.

Hebaixu Wang and Jiayi Ma. Deep adaptive unfolded network via spatial morphology stripping and spectral filtration for pan-sharpening. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10730–10740, 2025.

Hebaixu Wang, Meiqi Gong, Xiaoguang Mei, Hao Zhang, and Jiayi Ma. Deep unfolded network with intrinsic supervision for pan-sharpening. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 5419–5426, 2024a. doi: 10.1609/aaai.v38i6.28350.

Wu Wang, Liang-Jian Deng, Ran Ran, and Gemine Vivone. A general paradigm with detail-preserving conditional invertible network for image fusion. *International Journal of Computer Vision*, 132(4):1029–1054, 2024b. doi: 10.1007/s11263-023-01924-5.

Xueyang Wang, Zhixin Zheng, Jiandong Shao, Yule Duan, and Liang-Jian Deng. Adaptive rectangular convolution for remote sensing pansharpening. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17872–17881, 2025.

David P. Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science*, 10(1–2):1–157, 2014. doi: 10.1561/0400000060.

Zhong-Cheng Wu, Ting-Zhu Huang, Liang-Jian Deng, Jie Huang, Jocelyn Chanussot, and Gemine Vivone. LRTCFPan: Low-rank tensor completion based framework for pansharpening. *IEEE Transactions on Image Processing*, 32:1640–1655, 2023. doi: 10.1109/TIP.2023.3247165.

Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A Nyström-based algorithm for approximating self-attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 14138–14148, 2021. doi: 10.1609/aaai.v35i16.17664.

Junfeng Yang, Xueyang Fu, Yuwen Hu, Yue Huang, Xinghao Ding, and John Paisley. Pannet: A deep network architecture for pan-sharpening. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1753–1761, 2017. doi: 10.1109/ICCV.2017.193.

Roberta H. Yuhas, Alexander F. H. Goetz, and Joseph W. Boardman. Discrimination among semi-arid landscape endmembers using the spectral angle mapper algorithm. In *Summaries of the Third Annual JPL Airborne Geoscience Workshop*, volume 1, pp. 147–149, 1992.

Tian-Jing Zhang, Liang-Jian Deng, Ting-Zhu Huang, Jocelyn Chanussot, and Gemine Vivone. A triple-double convolutional neural network for panchromatic sharpening. *IEEE Transactions on Neural Networks and Learning Systems*, 34(11):9088–9101, 2023. doi: 10.1109/TNNLS.2022.3155655.

Chen-Yu Zhao, Tian-Jing Zhang, Ran Ran, Zhi-Xuan Chen, and Liang-Jian Deng. LGPConv: Learnable gaussian perturbation convolution for lightweight pansharpening. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pp. 4647–4655, 2023. doi: 10.24963/ijcai.2023/517.

Huanyu Zhou, Qingjie Liu, and Yunhong Wang. PanFormer: A Transformer based model for pan-sharpening. In *2022 IEEE International Conference on Multimedia and Expo*, pp. 1–6, 2022. doi: 10.1109/ICME52920.2022.9859770.

A GF2-TO-QB ZERO-SHOT TRANSFER

To complement the WV3-to-WV2 evaluation in the main paper, we consider a second cross-sensor setting in which each model is trained on GF2 and directly evaluated on QB without fine-tuning or deployment-time adaptation. Both sensors provide four multispectral bands, but their spatial and spectral responses differ. This protocol therefore isolates robustness to sensor shift without introducing a mismatch in the number of input bands. Table 5 reports the completed reduced-resolution (RR) and full-resolution (FR) reruns under the same metric implementations used in the main experiments.

Table 5: GF2-to-QB zero-shot transfer. All available models use GF2-trained weights and are evaluated directly on QB. Best results are **bolded**, and second-best results are underlined.

	Reduced resolution			Full resolution		
Method	PSNR↑	ERGAS↓	SAM↓	HQNR↑	$D_\lambda \downarrow$	$D_s \downarrow$
TDNet	25.36	15.93	18.51	0.475	0.423	0.171
DISPNet	30.11	9.31	8.15	0.878	0.070	0.056
UTeRM	29.52	10.21	13.58	0.846	0.123	0.039
DCINN	31.35	8.08	7.51	**0.900**	0.057	0.046
WFA-Net	32.59	6.99	**7.05**	0.884	**0.046**	0.074
ARConv	31.49	7.78	7.38	0.858	0.112	**0.035**
BNNPan	17.69	38.85	28.99	0.060	0.849	0.588
DAPNet	29.28	10.29	9.94	0.777	0.127	0.109
OSSNet	**33.00**	**6.62**	7.13	0.889	0.048	0.066

Among all nine reruns, OSSNet achieves the best PSNR and ERGAS and the second-best SAM at reduced resolution. At full resolution, it ranks second in both HQNR and D_λ, while the best D_s is obtained by ARConv. These complementary rankings show that OSSNet transfers competitively from GF2 to QB, although no single method dominates every distortion measure. Together with WV3-to-WV2 transfer, this experiment broadens the evaluation of cross-sensor generalization beyond a single sensor pair.

B COMPARISON OF POLYNOMIAL BASES

We changed only the polynomial basis and kept the OSSNet architecture and training protocol fixed. Table 6 reports zero-shot WV2 PSNR, reduced-resolution WV3 metrics, and full-resolution WV3 HQNR.

Table 6: Polynomial-basis comparison. Best and second-best values are bold and underlined, respectively.

Basis	WV2-ZS PSNR↑	WV3 reduced resolution			WV3-FR HQNR↑
		PSNR↑	ERGAS↓	SAM↓	
Bernstein	30.12	37.21	2.24	3.30	0.962
Jacobi	29.86	37.11	2.26	3.32	0.963
Legendre	29.76	37.18	2.25	**3.29**	0.959
Power	29.72	**37.22**	2.24	**3.29**	0.959
Chebyshev	**30.24**	**37.22**	**2.23**	**3.29**	**0.970**

Chebyshev achieved the best WV2 PSNR, WV3 ERGAS, and WV3 HQNR, while matching the best WV3 PSNR and SAM. It therefore gave the most consistent performance across reconstruction, cross-sensor transfer, and full-resolution evaluation.

C Energy Retention and Subspace Roles

The main paper motivates compact sketching from the low-rank structure of multispectral features. However, the learned sketch operators are not both constrained to reproduce an SVD/PCA solution. We therefore separate two questions in this analysis: how much input energy a sketch retains, and how closely its learned row space matches the energy-optimal SVD/PCA subspace. This distinction allows SVD/PCA to serve as a precise analytical reference without treating it as the optimization target of CCS.

For a feature matrix $\mathbf{F}$ and a row-orthogonal sketch $\mathbf{Q}$, we measure the retained energy and its complementary projection error:

$$\mathcal{E}_{\text{keep}} = \frac{\|\mathbf{QF}\|_F^2}{\|\mathbf{F}\|_F^2}, \qquad \mathcal{E}_{\text{rec}} = 1 - \mathcal{E}_{\text{keep}}. \tag{21}$$

Among all rank-d orthogonal projectors, the exact SVD/PCA projector attains the maximum possible $\mathcal{E}_{\text{keep}}$ and therefore serves as the energy-optimal reference. It is not used as a supervised target for the learned CCS projection. We evaluated all 48 QuickBird (QB) test samples at the deployed ranks: $d = 12$ for CCS and $d = 1$ per 2×2 window for CSS.

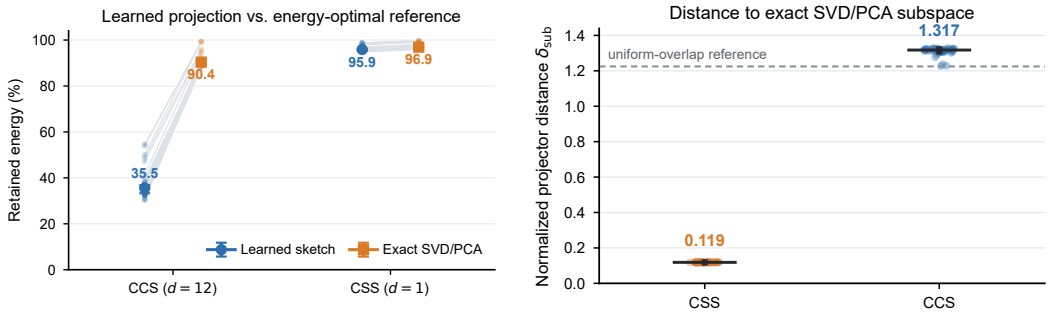
Subspace proximity. Energy retention alone does not determine whether two projectors select the same subspace. Let $\mathbf{U}_d$ contain the exact top-d SVD directions. We therefore measure

$$\delta_{\text{sub}} = \frac{\|\mathbf{Q}^\top\mathbf{Q} - \mathbf{U}_d\mathbf{U}_d^\top\|_F}{\|\mathbf{U}_d\mathbf{U}_d^\top\|_F}. \tag{22}$$

Because $\mathbf{QQ}^\top = \mathbf{I}_d$, $\mathbf{Q}^\top\mathbf{Q}$ is the orthogonal projector onto the learned row space of $\mathbf{Q}$. Likewise, $\mathbf{U}_d\mathbf{U}_d^\top$ projects onto the exact top-d SVD/PCA subspace. Comparing the projectors makes the measure invariant to the choice, sign, and rotation of the bases within each subspace. In terms of the principal angles $\{\theta_i\}_{i=1}^d$ between the two subspaces,

$$\delta_{\text{sub}}^2 = \frac{2}{d}\sum_{i=1}^{d} \sin^2\theta_i. \tag{23}$$

Hence, $\delta_{\text{sub}} = 0$ indicates identical subspaces, whereas $\delta_{\text{sub}} = \sqrt{2}$ corresponds to mutually orthogonal subspaces.



(a) Retained energy.

(b) Distance to the energy-optimal subspace.

Figure 4: Energy and subspace comparisons on 48 QB samples. (a) Retained energy at the deployed CCS and CSS ranks. (b) Normalized distance to the exact energy-optimal subspace; zero denotes identical projectors and $\sqrt{2}$ orthogonal projectors. Small points are image-level statistics; large markers and bars denote medians and interquartile ranges.

Figure 4a shows the different energy-retention roles of the two sketches. CSS retained 95.89% of local patch energy, close to exact local SVD (96.90%). CCS retained 35.53%, whereas the energy-optimal projector retained 90.38%. The median subspace distance was 0.119 for CSS and 1.317 for CCS (Fig. 4b). Thus, CSS closely followed the dominant local spatial direction, whereas CCS

learned a channel subspace that was geometrically distinct from PCA. The corresponding median CCS–PCA subspace overlap was only 0.132. These measurements distinguish the roles of the two sketches rather than ranking them against a shared learning objective. In particular, the lower CCS retention does not represent equivalent information loss in the final reconstruction. It measures the energy routed through the compact processing branch, while the complementary residual remains explicitly available through nullspace compensation.

How Does CSS Respond to Local Texture? CSS predicts a spatial sketch $\mathbf{Q}_{\mathrm{s}} \in \mathbb{R}^{1\times 4}$ for each 2×2 patch. If these directions remain close to $\mathbf{Q}_{\mathrm{avg}} = \frac{1}{2}[1, 1, 1, 1]$, CSS behaves similarly to fixed average pooling. We therefore examine whether $\mathbf{Q}_{\mathrm{s}}$ changes as the local texture becomes stronger.

We first measure patch texture independently of $\mathbf{Q}_{\mathrm{s}}$. For a patch $\mathbf{X} \in \mathbb{R}^{C\times 4}$, let

$$\bar{X}_c = \frac{1}{4}\sum_{p=1}^{4} X_{c,p}, \qquad \tau(\mathbf{X}) = \frac{\sum_{c=1}^{C}\sum_{p=1}^{4}(X_{c,p} - \bar{X}_c)^2}{\sum_{c=1}^{C}\sum_{p=1}^{4} X_{c,p}^2}. \tag{24}$$

A small $\tau(\mathbf{X})$ indicates a spatially smooth patch, whereas a large value indicates stronger local variation. Within each image and CSS stage, we rank the patches by $\tau(\mathbf{X})$ and divide them into four equally sized texture groups.

For each group, we measure how far the learned direction departs from fixed average pooling:

$$\delta_{\mathrm{avg}} = \left\|\mathbf{Q}_{\mathrm{s}}^{\mathsf{T}}\mathbf{Q}_{\mathrm{s}} - \mathbf{Q}_{\mathrm{avg}}^{\mathsf{T}}\mathbf{Q}_{\mathrm{avg}}\right\|_F = \sqrt{2 - 2\left|\mathbf{Q}_{\mathrm{s}}\mathbf{Q}_{\mathrm{avg}}^{\mathsf{T}}\right|^2}. \tag{25}$$

We also compute $|\mathbf{Q}_{\mathrm{s}}\mathbf{u}_1|^2$, where $\mathbf{u}_1$ is the dominant local SVD direction, to determine whether the adapted CSS direction continues to preserve the principal spatial mode.

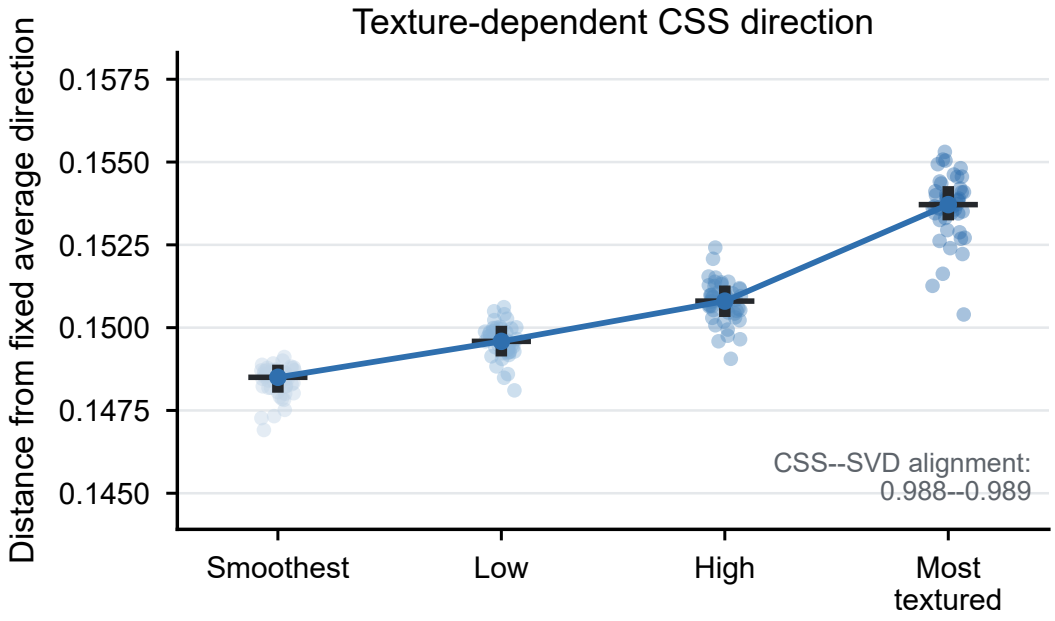


Figure 5: Texture-dependent CSS behavior on 48 QB images. Patches are divided into four groups according to $\tau(\mathbf{X})$. The vertical axis reports the projector distance δ_{avg} from fixed average pooling. Points are image-level statistics; bars denote medians and interquartile ranges.

Figure 5 shows that the median δ_{avg} increased from 0.149 for the smoothest patches to 0.154 for the most textured patches. Meanwhile, the CSS–SVD alignment remained between 0.988 and 0.989. CSS therefore adapts conservatively but consistently to local texture while preserving the dominant spatial direction.

D What Does CCS Learn Beyond PCA?

The preceding results reveal different roles for the two sketches. CSS closely follows the dominant local spatial direction. In contrast, CCS has a subspace distance of 1.317 from the rank-d PCA reference and therefore does not recover the variance-maximizing top-d channel subspace. This distinction does not contradict the low-rank motivation for compact computation. Rather, it motivates examining how CCS allocates its limited channel capacity and what information remains outside the selected subspace.

PCA maximizes retained input energy, whereas CCS is trained jointly with the pansharpening objective. Although polynomial matrix functions reshape a spectrum without explicit decomposition (Higham, 2008; Defferrard et al., 2016), the final CCS projection also depends on learned anchors, coefficients, and row orthogonalization. We therefore characterize CCS by its selection strength over covariance eigendirections. Let the centered bottleneck covariance be

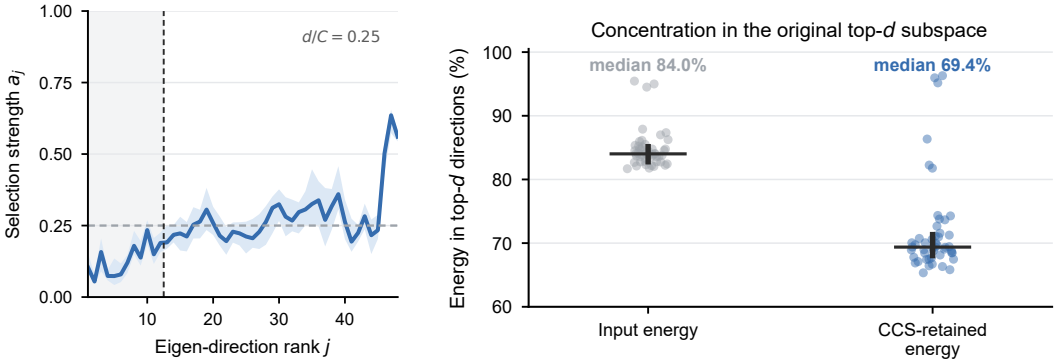
$$\mathbf{G}_c = \frac{1}{N_s}(\mathbf{F}_2^{k+1} - \overline{\mathbf{F}}_2^{k+1})(\mathbf{F}_2^{k+1} - \overline{\mathbf{F}}_2^{k+1})^\mathsf{T} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\mathsf{T}. \tag{26}$$

For eigen-direction $\mathbf{v}_j$, define

$$a_j = \|\mathbf{Q}_c\mathbf{v}_j\|_2^2, \qquad \sum_{j=1}^{C} a_j = d. \tag{27}$$

$$\mathrm{tr}(\mathbf{Q}_c\mathbf{G}_c\mathbf{Q}_c^\mathsf{T}) = \sum_{j=1}^{C} \lambda_j a_j. \tag{28}$$

Here, $a_j \in [0,1]$ is the selection strength: $a_j = 1$ means that $\mathbf{v}_j$ lies entirely in the learned row space, whereas $a_j = 0$ means that it is excluded. The product $\lambda_j a_j$ is the energy retained from direction j.



(a) Selection strength.

(b) Top-d energy concentration.

Figure 6: CCS selection relative to the covariance eigenspectrum on 48 QB samples. (a) Learned selection strength; the line and band denote the median and interquartile range, the vertical line marks $d = 12$, and d/C is the uniform-subspace reference. (b) Energy concentration within the original top-d eigendirections; points are samples and bars mark medians and interquartile ranges.

Figure 6a shows that CCS is not a binary top-d selector. It assigns limited strength to several leading directions while increasing the contribution of weaker ones. The original top-d directions contained 84.0% of input energy but only 69.4% of the energy retained by CCS (Fig. 6b). Together with the large CCS–PCA distance, this pattern shows that CCS distributes capacity beyond the dominant PCA directions. Because CCS is optimized jointly with the reconstruction network, this behavior is consistent with task-adaptive routing rather than variance-maximizing truncation. It does not imply that every selected low-energy covariance direction is useful in isolation or spatially high-frequency.

Why Does CCS Differ from PCA? The covariance $\mathbf{G}_c$ ranks centered spatial variation, whereas the CCS projection is applied to the uncentered bottleneck feature. To separate these quantities, we write

$$\mathbf{F}_2^{k+1} = \boldsymbol{\mu}\mathbf{1}^\mathsf{T} + \mathbf{F}_{2,\mathrm{c}}^{k+1}, \qquad \mathbf{F}_{2,\mathrm{c}}^{k+1}\mathbf{1} = \mathbf{0}, \tag{29}$$

where $\boldsymbol{\mu}$ is the per-channel spatial mean. Across the 48 QB samples, the mean and centered components contained median energy shares of 44.64% and 55.36%, respectively. CCS retained 61.67% of the mean component but only 13.46% of the centered variation. Their combination accounts for the observed 35.53% total retention.

This decomposition also explains why some low-variance covariance directions receive substantial selection strength. The leading 12 eigenvectors of $\mathbf{G}_c$ contained 31.70% of the mean-component energy, whereas the lowest 12 contained 41.67%. Retaining the spatially stable channel mean therefore need not align CCS with the dominant centered-covariance subspace. This accounts for part of the CCS–PCA distance without implying that CCS selects low-energy noise. Across all 1,128 sample pairs, the median normalized distance between CCS projectors was 0.047, with a maximum of 0.202. The input-dependent adjustment is therefore present but moderate. Overall, CCS behaves as a globally task-optimized and covariance-conditioned routing operator, rather than a sample-wise estimator of the PCA subspace.

What Does the Complementary Residual Contain? The spectral analysis above shows that CCS retains only part of the feature energy. We next test whether the complementary residual is unstructured redundancy or information that should remain available to the reconstruction network. Let

$$\mathbf{Z}_\mathrm{c} = \mathbf{Q}_\mathrm{c}\mathbf{F}_2^{k+1}, \qquad \mathbf{N}_\mathrm{c} = (\mathbf{I}_C - \mathbf{Q}_\mathrm{c}^\mathsf{T}\mathbf{Q}_\mathrm{c})\mathbf{F}_2^{k+1}. \tag{30}$$

By construction, $\mathbf{F}_2^{k+1} = \mathbf{Q}_\mathrm{c}^\mathsf{T}\mathbf{Z}_\mathrm{c} + \mathbf{N}_\mathrm{c}$. The measured near-row-orthogonality makes $\mathbf{N}_\mathrm{c}$ an approximately orthogonal complementary residual. We quantify the fraction of total and high-frequency input energy carried by this residual as

$$\rho_\mathrm{all} = \frac{\|\mathbf{N}_\mathrm{c}\|_F^2}{\|\mathbf{F}_2^{k+1}\|_F^2}, \qquad \rho_\mathrm{HF} = \frac{\|\mathcal{F}_\mathrm{HP}(\mathbf{N}_\mathrm{c})\|_F^2}{\|\mathcal{F}_\mathrm{HP}(\mathbf{F}_2^{k+1})\|_F^2}, \tag{31}$$

where $\mathcal{F}_\mathrm{HP}$ is the fixed 3×3 local-average high-pass operator used only for this diagnostic analysis. After sketch processing, OSSNet fuses the lifted response with this residual:

$$\mathbf{U}_2 = \mathrm{Fuse}_2\big(\mathbf{Q}_\mathrm{c}^\mathsf{T}\mathcal{B}(\mathbf{Z}_\mathrm{c}), \mathbf{N}_\mathrm{c}\big)\,, \tag{32}$$

where Fuse_2 is implemented by concatenation followed by a 1×1 convolution.

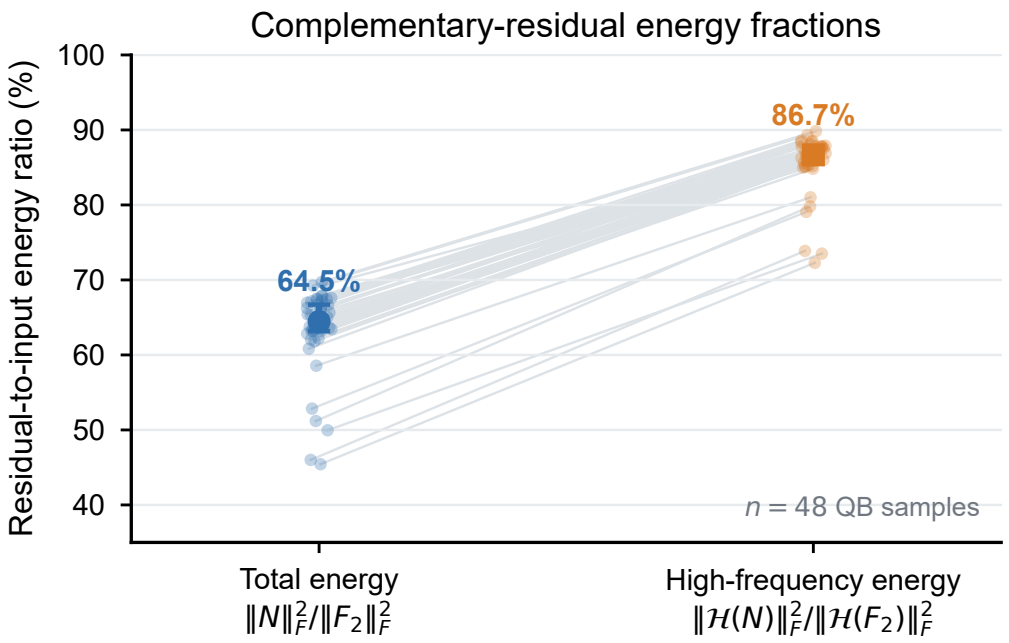
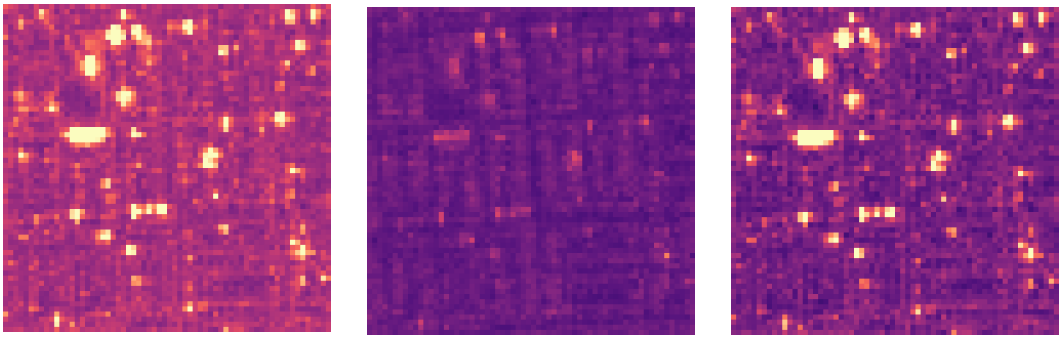


Figure 7: Residual-to-input energy ratios on 48 QB samples. The two ratios are $\rho_{\text{all}} = \|\mathbf{N}_{\text{c}}\|_F^2/\|\mathbf{F}_2^{k+1}\|_F^2$ and $\rho_{\text{HF}} = \|\mathcal{F}_{\text{HP}}(\mathbf{N}_{\text{c}})\|_F^2/\|\mathcal{F}_{\text{HP}}(\mathbf{F}_2^{k+1})\|_F^2$. Large markers and bars denote medians and interquartile ranges across samples.

Across the 48 samples, the median ρ_{all} was 64.5%: for a typical sample, $\mathbf{N}_{\text{c}}$ carried 64.5% of the total energy in $\mathbf{F}_2^{k+1}$. The median ρ_{HF} was 86.7%, meaning that $\mathbf{N}_{\text{c}}$ carried most of the high-frequency energy measured in $\mathbf{F}_2^{k+1}$ (Fig. 7).



(a) Input $\mathbf{F}_2^{k+1}$ (b) Projection (c) Complement $\mathbf{N}_{\text{c}}$

Figure 8: Same-scale channel-RMS maps for the sample whose ρ_{HF} is closest to the median over all 48 QB samples.

Under the common color scale in Fig. 8, the input map contains sparse bright responses superimposed on lower-amplitude spatial texture. The projection retains weaker, spatially coherent responses but strongly attenuates most localized peaks. In contrast, $\mathbf{N}_{\text{c}}$ preserves the locations and shapes of these peaks together with much of the fine-grained background variation. This spatial organization indicates that $\mathbf{N}_{\text{c}}$ contains structured complementary information rather than unstructured discarded noise. Its mixture of localized peaks and broader patterns also shows that it is not exclusively high-frequency.

Taken together, these observations refine rather than replace the low-rank motivation in the main paper. Classical SVD/PCA provides the energy-optimal reference for compact subspace processing. CSS behaves similarly to this reference at the local spatial scale, whereas CCS learns a different, task-optimized and covariance-conditioned channel row space. CCS is therefore best understood as an approximately row-orthogonal channel analysis–synthesis and routing operator, rather than a PCA or truncated-SVD approximation. The compact branch preferentially processes the stable channel component. Nullspace compensation preserves most centered variation and high-frequency energy for subsequent reconstruction.

E Implementation Details

OSSNet was implemented in PyTorch and trained on Ubuntu 24.04.2 LTS using single-GPU jobs on an NVIDIA RTX 5070 (12 GB) or RTX 5060 Ti (16 GB). The environment used Python 3.10.20, PyTorch 2.7.0, CUDA 12.8, cuDNN 9.7.1, and NVIDIA driver 575.57.08. Table 7 lists the training settings. The empirical inference latency and peak GPU memory reported in Table 2 were measured on a single NVIDIA RTX 5060 Ti.

Table 7: OSSNet training and efficiency-measurement configuration.

Setting	Value
Optimizer	Adam
Initial learning rate	1×10^{-3}
Adam coefficients	$\beta_1 = 0.9,\ \beta_2 = 0.999$
Weight decay	0
Training epochs	360
Batch size	8
Learning-rate schedule	MultiStepLR
Decay milestones	$[100, 200, 300, 330]$
Decay factor	0.5
Training patch size	64×64
Latent feature width	$C = 48$
Primary CCS rank	$d = 12$
Chebyshev polynomial order	$K = 3$
CSS covariance channels	$C' = 16$
Newton–Schulz iterations	5
Spatial augmentation	Random flipping and rotation
Reconstruction loss	ℓ_1 loss
Orthogonality weight	$\lambda_{\mathrm{orth}} = 10^{-4}$
Latency protocol	RTX 5060 Ti; FP32; batch size 1; $\mathbf{I}_{\mathrm{LMS}} \in \mathbb{R}^{4\times256\times256}$ and $\mathbf{I}_{\mathrm{P}} \in \mathbb{R}^{1\times256\times256}$; 50 warm-up and 100 timed forward passes; CUDA events with synchronization; mean per-image latency