Training and Inference within 1 Second – Tackle Cross-Sensor Degradation of Real-World Pansharpening with Efficient Residual Feature Tailoring

Tianyu Xin[1*], **Jin-Liang Xiao**[1*], **Zeyu Xia**[1*], **Shan Yin**[1], **Liang-Jian Deng**[1†]

[1]University of Electronic Science and Technology of China
tyxin@std.uestc.edu.cn, jinliang_xiao@163.com, zeyuxia42@std.uestc.edu.cn, yins@std.uestc.edu.cn, liangjian.deng@uestc.edu.cn

Abstract

Deep learning methods for pansharpening have advanced rapidly, yet models pretrained on data from a specific sensor often generalize poorly to data from other sensors. Existing methods to tackle such cross-sensor degradation include retraining model or zero-shot methods, but they are highly time-consuming or even need extra training data. To address these challenges, our method first performs modular decomposition on deep learning-based pansharpening models, revealing a general yet critical interface where high-dimensional fused features begin mapping to the channel space of the final image. A Feature Tailor is then integrated at this interface to address cross-sensor degradation at the feature level, and is trained efficiently with physics-aware unsupervised losses. Moreover, our method operates in a patch-wise manner, training on partial patches and performing parallel inference on all patches to boost efficiency. Our method offers two key advantages: (1) *Improved Generalization Ability*: it significantly enhance performance in cross-sensor cases. (2) *Low Generalization Cost*: it achieves sub-second training and inference, requiring only partial test inputs and no external data, whereas prior methods often take minutes or even hours. Experiments on the real-world data from multiple datasets demonstrate that our method achieves state-of-the-art quality and efficiency in tackling cross-sensor degradation. For example, training and inference of $512 \times 512 \times 8$ image within *0.2 seconds* and $4000 \times 4000 \times 8$ image within *3 seconds* at the fastest setting on a commonly used RTX 3090 GPU, which is over 100 times faster than zero-shot methods.

Code — https://github.com/TorwnexialX/ERFT
Extended version — https://arxiv.org/abs/2508.07369

Introduction

Remote sensing images are widely utilized in various fields. Due to hardware constraints, satellites typically capture low-resolution multispectral (LRMS) images and high-resolution panchromatic (PAN) images separately. And the target of pansharpening is to fuse an LRMS image and a PAN image to a high-resolution multispectral (HRMS) image that preserves both spectral and spatial information.

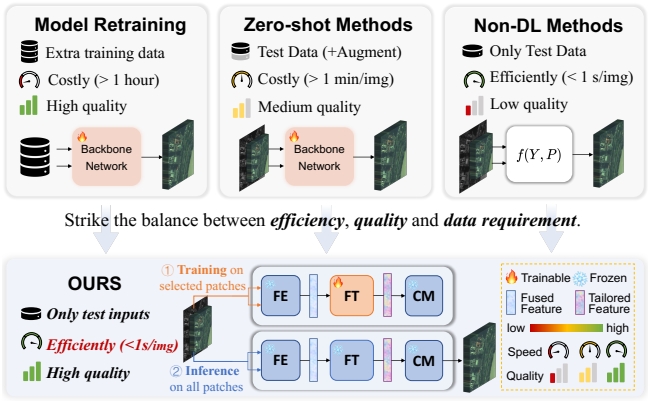

Figure 1: Comparison of different cross-sensor pansharpening methods. Existing approaches (upper panel) struggle to balance efficiency, quality, and data requirement. In contrast, our method (lower panel) achieves state-of-the-art performance with sub-second efficiency, requires only test-time inputs, and fully leverages pretrained model capabilities.

Pansharpening methods are broadly classified into four categories: component substitution, multi-resolution analysis, variational optimization, and deep learning (DL) approaches (Cao et al. 2025b,a). The first three categories are traditional methods that process inputs via mathematical transformations without model training, but their inability to capture nonlinear PAN–HRMS relationships limits fusion quality (Xiao et al. 2022, 2023). In contrast, deep learning methods (Li et al. 2023; Liu et al. 2024) leverage excellent feature extraction capability of neural networks, such as convolutional neural network (CNN)-based models (He et al. 2019; Cao et al. 2021; Wang et al. 2024), Transformer-based models (Li et al. 2024; Zhang and Ma 2021), and Diffusion-based methods (Cao et al. 2024b; Zhong et al. 2024).

While DL methods have achieved impressive results, the growing diversity of satellite sensors makes *cross-sensor generalization increasingly important* for existing pansharpening models. However, *current DL methods often suffer from cross-sensor degradation*, where models pretrained on data from a specific sensor generalize poorly to data from others. As shown in Figure 1 (upper panel), typical strategies

[*]These authors contributed equally.
[†]Corresponding author.

addressing the issue include model retraining or zero-shot methods, but they are highly time-consuming or even need extra training data. Zero-shot methods (Rui et al. 2024; Cao et al. 2024a) usually take minutes to process only one input pair (512×512 PAN), let alone model retraining that takes hours. And although traditional methods are more efficient, their fusion quality remains significantly lower than that of DL methods. More importantly, most existing methods often disrupt the pretrained model weights, failing to fully leverage the rich representational capacity already embedded in them. *In sum, current methods for addressing cross-sensor degradation struggle to balance the trade-off between quality, efficiency, and data requirement, while also underutilizing the capabilities of existing pretrained models.*

Therefore, two critical challenges in tackling cross-sensor degradation for real-world pansharpening remain underexplored: 1) *minimizing generalization cost by reducing processing time and avoiding the need for extra training data*; 2) *preserving the pretrained model's existing capabilities while ensuring improved generalization to unseen sensors*. To address the challenges, we propose a plug-and-play solution to tackle cross-sensor degradation with sub-second efficiency: the **Efficient Residual Feature Tailoring (ERFT)** pipeline. It begins by modularly decomposing pansharpening models into two general components: Feature Extractor (FE) and Channel Mapper (CM). The FE generates high-dimensional fused feature from the LRMS-PAN input with various neural network architectures. And the CM maps the feature from feature space to channel space to generate the high resolution output image, mostly with a shallow CNN structure. Then we integrates a **Feature Tailor (FT)** between frozen FE and CM to address cross-sensor degradation of pretrained model at feature level while leveraging capabilities of pretrained models. To maximize efficiency and ensure structural compatibility, the FT also employs a CNN structure, which is trained with physics-aware unsupervised losses to preserve spectral and spatial fidelity. To further improve runtime efficiency, our method operates in a patch-wise manner during both training and inference. Specifically, the LRMS-PAN input is first partitioned into multiple patches. A randomly selected subset is used to train the FT, while keeping the pretrained FE and CM frozen. Then all modules (FE, FT, and CM) are fixed to perform parallel inference on all patches. Finally, the predicted HRMS patches are stitched together to form the final high-resolution output.

Experiments on real-world data from sensors with different band setting again prove that our method tackles cross-sensor degradation with *state-of-the-art (SOTA) quality and efficiency*. Moreover, our method could process input images scaled up to megapixel level (images with over one million pixels). In detail, with a commonly used RTX 3090 GPU, our method can complete pansharpening of $512 \times 512 \times 8$ image within *0.2 seconds* and $4000 \times 4000 \times 8$ within *3 seconds* at the fastest setting, which is *over 100 times faster* than zero-shot methods and outperform latest pansharpening results of 2025. To the best of our knowledge, our method is the *first* to tackle cross-sensor degradation with the *sub-second efficiency* while achieving SOTA fusion quality.

The main contributions are summarized as follows:

- We propose the Efficient Residual Feature Tailoring (ERFT) to address the efficiency challenge of cross-sensor generalization. It avoids the need for extra training data and achieves training and inference within a sub-second runtime that has not been attained in this domain.
- Our method offers a novel plug-and-play solution for cross-sensor degradation in real-world pansharpening. It avoids modifying the pretrained model parameters, thereby preserving their existing capabilities while enabling improved generalization to unseen sensors.
- Extensive experiments demonstrate that our method achieves state-of-the-art results on both in-sensor and cross-sensor datasets with sub-second efficiency. Its ability to process megapixel-scale inputs within 3 seconds further showcases its scalability and practical potential.

Related Works

Deep Learning-based Pansharpening

Deep learning (DL)-based methods have significantly advanced the task of pansharpening by leveraging diverse neural network architectures to learn complex nonlinear mappings and extract rich spatial-spectral representations. CNN-based models such as PanNet (Yang et al. 2017), DiCNN (He et al. 2019), and FusionNet (Deng et al. 2020) utilize convolutional layers to capture hierarchical features, and have demonstrated strong performance in enhancing spatial detail while maintaining spectral fidelity. Building on these foundations, Transformer-based models (Zhou, Liu, and Wang 2022; Li et al. 2024) and diffusion-based approaches (Meng et al. 2023; Zhang et al. 2024) introduce more expressive architectures that can better model long-range dependencies and complex input distributions. These models enable deeper and more flexible nonlinear transformations, leading to more accurate and robust spatial-spectral fusion. *Despite their impressive results, most deep learning-based methods suffer from evident cross-sensor degradation.* Specifically, models pretrained on data from a specific sensor often generalize poorly to data from other sensors due to data distribution shifts in spectral and spatial characteristics. This significantly limits their practical applicability in real-world scenarios involving diverse satellite sources.

Cross-Sensor Application of Pansharpening

Applying pansharpening models across different satellite sensors presents significant challenges due to variations in spectral responses and spatial characteristics. To address this, existing methods typically follow three main strategies:

(1) **Model Retraining**: DL models are retrained on data from the target sensor. While effective, this approach requires substantial training time (often several hours) and access to extra training data on the new sensor, and often fails to fully leverage the pretrained model's existing knowledge.

(2) **Zero-Shot Methods**: These methods (Rui et al. 2024; Cao et al. 2024a) optimize pansharpening models during inference without requiring supervision. They eliminate the need for extra training data and maintain relatively high fusion quality. However, they typically require several minutes

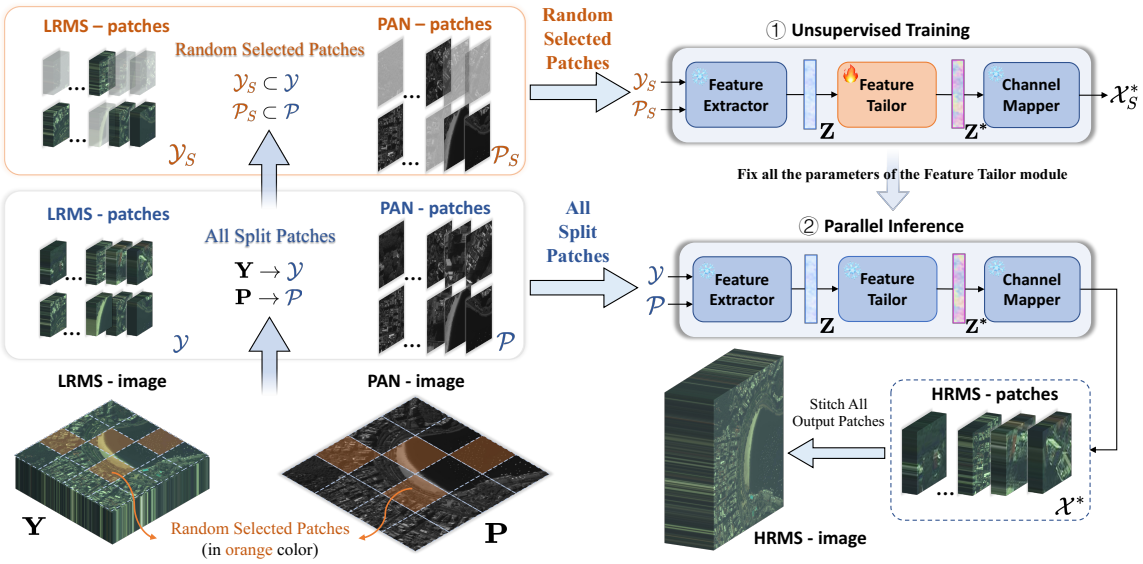

Figure 2: Our Efficient Residual Feature Tailoring pipeline is conducted in a patch-wise manner. Specifically, (1) Random selected patches are selected for unsupervised training of the Feature Tailor, enabling the feature-level adjustments; (2) Parallel inference is conducted on all split patches, whose resulting HRMS patches are stitched together to form the final HRMS image.

to process a single LRMS-PAN input, highlighting the efficiency challenges of cross-sensor generalization.

(3) **Traditional Methods**: Traditional non-DL approaches such as Component Substitution (CS) (Choi, Yu, and Kim 2011; Vivone 2019a), Multiresolution Analysis (MRA) (Vivone et al. 2014; Vivone, Restaino, and Chanussot 2018a), and Variational Optimization (VO) (Fu et al. 2019; Tian et al. 2022) are efficient and require only a single PAN-LRMS pair as the input, but their fusion quality lags significantly behind modern deep learning methods.

In summary, existing strategies struggle to balance the trade-off between quality, efficiency, and data requirements, highlighting the challenge of efficient cross-sensor generalization without requiring additional training data.

Methodology

As shown in Figure 2, we propose the plug-and-play pipeline named **Efficient Residual Feature Tailoring (ERFT)** to efficiently address cross-sensor degradation in pansharpening. Our method comprises three components: (1) residual feature tailoring for cross-sensor generalization; (2) patch-wise training and inference for improved efficiency; (3) physics-aware unsupervised losses for spatial and spectral fidelity.

Residual Feature Tailoring

In this section, we detail the design of our **Feature Tailor (FT)** module, including both its placement and structure.

(I) Tailoring Position Design via Modular Decomposition DL-based pansharpening models generally take an LRMS image $\mathbf{Y}$ and a PAN image $\mathbf{P}$ as inputs, and produces a C-channel HRMS output $\hat{\mathbf{X}}$, which can be described as:

$$\hat{\mathbf{X}} = \mathcal{F}(\mathbf{Y}, \mathbf{P}; \theta), \quad \hat{\mathbf{X}} \in \mathbb{R}^{C \times H \times W}, \tag{1}$$

where $\mathcal{F}$ denotes the backbone network and θ its parameters.

Despite architectural variations, most existing methods (Yang et al. 2017; He et al. 2019; Liang et al. 2022) share a common modular pattern: an early-stage feature extractor $\mathcal{F}_1$ that encodes and fuses input information, and a late-stage channel mapper $\mathcal{F}_2$ that projects the fused features back into the output space. This decomposition yields:

$$\mathcal{F} = \mathcal{F}_2 \circ \mathcal{F}_1, \quad \theta = \theta_1 \cup \theta_2, \tag{2}$$

where θ_1, θ_2 are the parameters of $\mathcal{F}_1$ and $\mathcal{F}_2$, respectively.

The feature extractor $\mathcal{F}_1$ transforms inputs into a high-dimensional representation $\mathbf{Z}$ with S latent dimensions:

$$\mathbf{Z} = \mathcal{F}_1(\mathbf{Y}, \mathbf{P}; \theta_1), \quad \mathbf{Z} \in \mathbb{R}^{S \times H \times W}. \tag{3}$$

The channel mapping stage then reconstructs the HRMS image $\hat{\mathbf{X}}$ from $\mathbf{Z}$, typically with a residual shortcut from the upsampled LRMS image $\mathbf{Y}$ to maintain spectral fidelity:

$$\hat{\mathbf{X}} = \mathcal{F}_2(\mathbf{Z}; \theta_2) + \texttt{UpSample}(\mathbf{Y}). \tag{4}$$

We insert the **Feature Tailor (FT)** module at the junction between $\mathcal{F}_1$ and $\mathcal{F}_2$. This critical interface carries the fused feature $\mathbf{Z}$, which is rich in information from both modalities yet unconstrained by the output space, thus serving as a favorable target for cross-sensor generalization. This choice is also supported by prior studies (Dou et al. 2019; Yosinski et al. 2014), which show that intermediate features possess greater transferability and representational flexibility than shallow or output-level features. Moreover, our experiments further validate this design: as shown in Figure 3, feature-space adjustments by placing the FT before the channel mapper (pre-CM) consistently outperform output-space adjustments by placing the FT after the channel mapper (post-CM) and the baseline, highlighting this interface as an ideal and effective position for cross-sensor generalization.

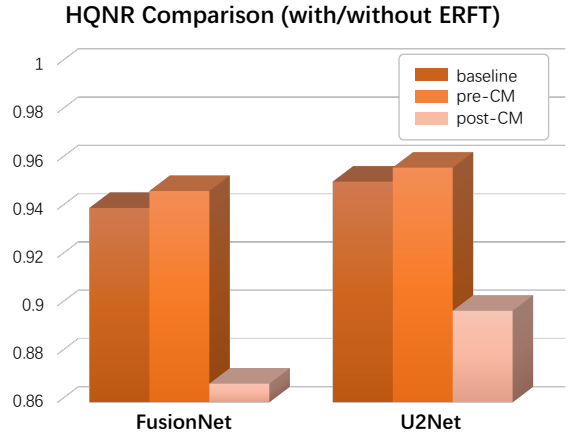

Figure 3: HQNR comparison for FusionNet and U2Net on WV3 dataset under three FT placement strategies: FT not inserted (Baseline), FT inserted before channel mapping (pre-CM), and FT inserted after channel mapping (post-CM). The pre-CM configuration consistently outperforms the others, validating the effectiveness of feature-level adjustments.

(II) Structural Design of the Feature Tailor The Feature Tailor (FT) module is implemented as a residual block with shallow CNN. Positioned at the intermediate interface between the feature extractor $\mathcal{F}_1$ and the channel mapper$\mathcal{F}_2$, it operates on the fused latent representation$\mathbf{Z}$, which encodes rich spatial and spectral cues from both input modalities.

Instead of generating new representations from scratch, the FT module learns a residual adjustment $\Delta(\mathbf{Z})$ using unsupervised losses to adapt $\mathbf{Z}$ to the test-time distribution. Formally, the tailored feature $\mathbf{Z}^*$ is computed as:

$$\mathbf{Z}^* = \mathbf{Z} + \mathcal{G}(\mathbf{Z}; \phi), \quad \mathbf{Z}^* \in \mathbb{R}^{S \times H \times W}, \tag{5}$$

where $\mathcal{G}(\cdot)$ is a shallow CNN network with parameters ϕ.

The tailored feature $\mathbf{Z}^*$ is then passed to the frozen channel mapper $\mathcal{F}_2$ to reconstruct the final HRMS output:

$$\hat{\mathbf{X}}^* = \mathcal{F}_2(\mathbf{Z}^*; \theta_2) + \texttt{UpSample}(\mathbf{Y}), \tag{6}$$

where $\hat{\mathbf{X}}^* \in \mathbb{R}^{C \times H \times W}$ denotes the ERFT-enhanced output.

This residual design is motivated by three key considerations. First, it allows the FT module to make precise and efficient adjustments to the latent features without disrupting the pretrained backbone. Second, using a shallow CNN ensures both fast optimization and structural continuity, as the following channel mapper $\mathcal{F}_2$ is also typically implemented as a shallow CNN. Finally, the design aligns with findings from existing works (Ilyas et al. 2019; von Kügelgen et al. 2025), which demonstrate that even small perturbations to intermediate features can significantly alter predictions, suggesting that learning residual feature corrections is an effective means of improving cross-sensor generalization.

Efficient Patch-wise Training and Inference

To improve runtime efficiency, we introduce a patch-wise strategy for both training and inference. As shown in Algorithm 1, rather than processing the entire image at once, each PAN–LRMS input pair is partitioned into multiple patches. A randomly selected subset of these patches is then used to train the FT module in an unsupervised manner. Once the training is completed, the FT parameters are fixed and subsequently applied to inference on all split patches. For clarity, we omit the outer loop over the training epochs for Lines 3 to 11 in Algorithm 1, though it is used in practice.

Algorithm 1: Patch-wise Workflow of ERFT

Input: PAN image $\mathbf{P}$, LRMS image $\mathbf{Y}$, backbone net $\mathcal{F}$
Output: HRMS output image $\hat{\mathbf{X}}^*$

1: Split $\mathbf{Y}, \mathbf{P}$ into N patches: $\{(\mathbf{Y}_i, \mathbf{P}_i)\}_{i=1}^{N}$
2: Randomly select M training patches: $\mathcal{T} \subset \{1, \ldots, N\}$
3: Initialize loss: $\mathcal{L} \leftarrow 0$
4: **for** each $i \in \mathcal{T}$ **do** ▷ Training on selected patches
5: $\quad \mathbf{Z}_i \leftarrow \mathcal{F}_1(\mathbf{Y}_i, \mathbf{P}_i)$
6: $\quad \mathbf{Z}_i^* \leftarrow \mathcal{G}(\mathbf{Z}_i) + \mathbf{Z}_i$ ▷ Residual feature tailoring
7: $\quad \hat{\mathbf{X}}_i \leftarrow \mathcal{F}_2(\mathbf{Z}_i^*) + \mathrm{UpSample}(\mathbf{Y}_i)$
8: $\quad \hat{\mathbf{X}}_i^0 \leftarrow \mathcal{F}_2(\mathbf{Z}_i) + \mathrm{UpSample}(\mathbf{Y}_i)$
9: $\quad \mathcal{L} \leftarrow \mathcal{L} + \mathcal{L}_{\text{unsup}}(\hat{\mathbf{X}}_i, \hat{\mathbf{X}}_i^0, \mathbf{Y}_i, \mathbf{P}_i)$
10: **end for**
11: Update $\mathcal{G}$ using accumulated loss $\mathcal{L}$
12: Freeze $\mathcal{G}$
13: **for** each $i = 1$ to N **do** ▷ Inference on all patches
14: $\quad \mathbf{Z}_i \leftarrow \mathcal{F}_1(\mathbf{Y}_i, \mathbf{P}_i)$
15: $\quad \mathbf{Z}_i^* \leftarrow \mathcal{G}(\mathbf{Z}_i) + \mathbf{Z}_i$ ▷ Residual feature tailoring
16: $\quad \hat{\mathbf{X}}_i \leftarrow \mathcal{F}_2(\mathbf{Z}_i^*) + \mathrm{UpSample}(\mathbf{Y}_i)$
17: **end for**
18: Get the final output HRMS image $\hat{\mathbf{X}}^* \leftarrow \mathrm{Stitch}(\{\hat{\mathbf{X}}_i\})$

Architecture	Before	After	Speedup
CNN	$\mathcal{O}(HW)$	$\mathcal{O}(\frac{M}{B}hw)$	$\frac{N}{M}B$
Attention	$\mathcal{O}(H^2W^2)$	$\mathcal{O}(\frac{M}{B}h^2w^2)$	$\frac{N^2}{M}B$
CNN	$\mathcal{O}(HW)$	$\mathcal{O}(\frac{N}{B}hw)$	B
Attention	$\mathcal{O}(H^2W^2)$	$\mathcal{O}(\frac{N}{B}h^2w^2)$	$N \cdot B$

Table 1: Theoretical complexity and speedup of our patch-wise strategy for CNN and Attention architectures. "Before" denotes complexity of full-image processing; "After" denotes complexity of patch-wise processing. For training (upper block), only M out of N patches are used; for inference (lower block), all N patches are processed in parallel. Here, $H \times W$ is the input image size, $h \times w$ is the patch size.

Such patch-wise strategy for both training and inference greatly reduces the time complexity. By training the FT module on only a small subset of patches, it lowers the number of forward and backward passes, accelerating optimization. Moreover, since patches are processed independently, they can be grouped into batches and executed in parallel during both training and inference, further improving efficiency. This strategy is especially beneficial for backbone networks with superlinear time complexity, such as Transformers. The theoretical complexities and speedup

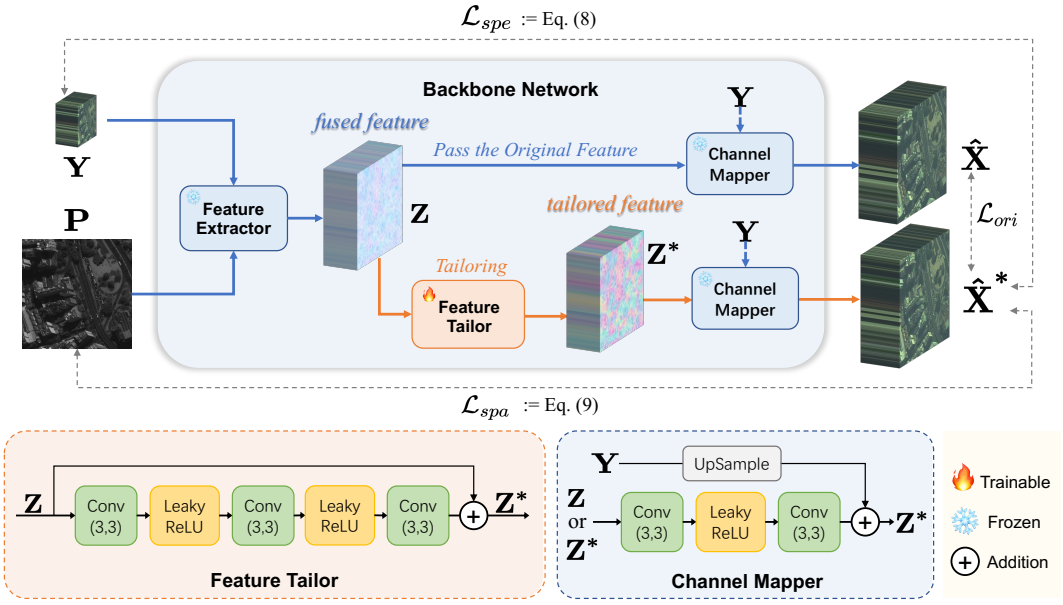

Figure 4: Detailed workflow of unsupervised training. The LRMS image $\mathbf{Y}$ and PAN image $\mathbf{P}$ are fed into the backbone network to extract high-dimensional latent features $\mathbf{Z}$, which are then refined by the FT module to produce tailored features $\mathbf{Z}^*$. Both $\mathbf{Z}$ and $\mathbf{Z}^*$ are passed through the channel mapping (CM) module to generate the original and tailored HRMS outputs, $\hat{\mathbf{X}}$ and $\hat{\mathbf{X}}^*$, respectively. These outputs are compared with the inputs to compute unsupervised losses to update the FT module.

gains of our patch-wise strategy for typical CNN-based and attention-based architectures are reported in Table 1, with formal derivations provided in the *extended version*.

Unsupervised FT Training Loss

To address cross-sensor degradation using only test inputs, we optimize our framework (Fig. 4) with the unsupervised loss from Eq.(7), which combines spectral ($\mathcal{L}_{spe}$), spatial ($\mathcal{L}_{spa}$), and original-output consistency ($\mathcal{L}_{ori}$) losses. Patch indices are omitted from the following formulas for clarity.

$$\mathcal{L}_{total} = \eta_1 \mathcal{L}_{spe} + \eta_2 \mathcal{L}_{spa} + \eta_3 \mathcal{L}_{ori}, \tag{7}$$

where η_1, η_2, and η_3 balance the contribution of each term.

(I) Spectral Loss ($\mathcal{L}_{spe}$) This term preserves spectral fidelity by encouraging the blurred and downsampled HRMS output $\hat{\mathbf{X}}^*$ to match the LRMS input $\mathbf{Y}$, where $\mathbf{B}$ denotes the modulation transfer function (MTF)-based blur kernel:

$$\mathcal{L}_{spe} = \|\text{DownSample}(\hat{\mathbf{X}}^*\mathbf{B}) - \mathbf{Y}\|_1. \tag{8}$$

(II) Spatial Loss ($\mathcal{L}_{spa}$) This term promotes spatial consistency by aligning the high-frequency details of $\hat{\mathbf{X}}^*$ and PAN $\mathbf{P}$, following method in (Wu et al. 2022). In Eq. (9), $\hat{\mathbf{P}}$ denotes the PAN broadcasted to C channels, $\circ$ is element-wise multiplication, and $\oslash$ is element-wise division:

$$\mathcal{L}_{spa} = \|\hat{\mathbf{X}}^* - \hat{\mathbf{X}}^*\mathbf{B} \circ (\hat{\mathbf{P}} \oslash \hat{\mathbf{P}}\mathbf{B})\|_1. \tag{9}$$

(III) Consistency Loss ($\mathcal{L}_{ori}$) This term prevents overfitting by regularizing the ERFT-enhanced output to remain close to the original prediction from the frozen backbone:

$$\mathcal{L}_{ori} = \|\hat{\mathbf{X}}^* - \hat{\mathbf{X}}\|_1. \tag{10}$$

Experiments

Experiment Settings

(I) Datasets and Metrics We conduct experiments on four datasets derived from the PanCollection [1]. These datasets are captured by WorldView-3 (WV-3), WorldView-2 (WV-2), QuickBird (QB), and Gaofen-2 (GF2), and consist of paired PAN and LRMS images, with each PAN image sized at 512×512 pixels. Our method is specifically designed for real-world pansharpening, which is of paramount practical significance. Accordingly, we primarily evaluate performance on real-world data using three widely adopted reference-free metrics: HQNR (Aiazzi et al. 2014), D_s, and D_λ. Notably, HQNR is derived from D_s and D_λ, providing an assessment of both spatial and spectral fidelity.

(II) Benchmarks We compare our method with several SOTA pansharpening approaches, including *six traditional methods*, BT-H (Lolli et al. 2017), C-BDSD (Garzelli 2014), BDSD-PC (Vivone 2019b), MTF-GLP (Aiazzi et al. 2006), MTF-GLP-FS (Vivone, Restaino, and Chanussot 2018b), and MF (Restaino et al. 2016); and *six DL-based methods*, FusionNet (Deng et al. 2020), U2Net (Peng et al. 2023), Ps-Dip (Rui et al. 2024), ZS-Pan (Cao et al. 2024a), Fusion-Mamba (Peng et al. 2024), and WFANet (Huang et al. 2025). Then we employ FusionNet (Deng et al. 2020) and U2Net (Peng et al. 2023) as our baseline networks, representing typical CNN and attention-based architectures, respectively.

(III) Implementation Details All experiments were conducted on a hardware setup comprising an NVIDIA RTX

[1] https://github.com/liangjiandeng/PanCollection

Method	QB → GF-2 (4-band sensor): Avg±std			WV-3 → WV-2 (8-band sensor): Avg±std		
	HQNR↑	$D_\lambda \downarrow$	$D_s \downarrow$	HQNR↑	$D_\lambda \downarrow$	$D_s \downarrow$
BT-H	0.7293±0.0253	0.1559±0.0239	0.1359±0.0192	0.8300±0.0430	0.0860±0.0301	0.0925±0.0208
C-BDSD	0.6996±0.0323	0.2149±0.0302	0.1091±0.0154	0.6956±0.0461	0.2253±0.0488	0.1019±0.0296
BDSD-PC	0.6762±0.0378	0.1971±0.0346	0.1578±0.0279	0.8286±0.0432	0.1413±0.0320	0.0356±0.0213
MTF-GLP	0.7151±0.0366	0.1592±0.0254	0.1495±0.0355	0.8549±0.0475	0.0582±0.0221	0.0930±0.0320
MTF-GLP-FS	0.7423±0.0338	0.1438±0.0251	0.1331±0.0291	0.8658±0.0415	0.0563±0.0212	0.0830±0.0260
MF	0.7817±0.0231	**0.0600±0.0260**	0.1470±0.0223	0.8508±0.0538	0.0704±0.0308	0.0857±0.0297
PsDip	0.7825±0.0325	0.1323±0.0286	0.0981±0.0256	0.8980±0.0226	**0.0385±0.0239**	0.0659±0.0158
ZS-Pan	0.8925±0.0240	0.0778±0.0182	**0.0323±0.0159**	0.9112±0.0336	0.0476±0.0270	0.0435±0.0130
FusionMamba	0.7627±0.0724	0.1627±0.0695	0.0893±0.0352	0.9022±0.0242	0.0572±0.0272	0.0429±0.0112
WFANet	0.7093±0.0450	0.1975±0.0497	0.1154±0.0382	**0.9128±0.0301**	0.0526±0.0302	0.0366±0.0054
FusionNet	0.8397±0.0509	0.1114±0.0500	0.0551±0.0137	0.8881±0.0213	0.0543±0.0273	0.0606±0.0162
U2Net	0.7089±0.0535	0.2209±0.0574	0.0909±0.0286	0.8706±0.0809	0.0936±0.0790	0.0400±0.0098
$\text{ERFT}_{\text{FusionNet}}$	**0.8970±0.0471**	0.0706±0.0382	0.0353±0.0145	0.9100±0.0255	0.0445±0.0214	0.0475±0.0160
$\text{ERFT}_{\text{U2Net}}$	0.8264±0.1175	0.0930±0.1279	0.0888±0.0021	**0.9128±0.0526**	0.0617±0.0494	**0.0274±0.0073**

Table 2: Performance comparison in cross-sensor scenarios on real-world datasets. "A → B" indicates that the model is pretrained on dataset A and tested on dataset B. All results are averaged over 20 test images. (**Bold**: best; Underline: second best)

3090 GPU with 24GB memory and Intel i9-12900 CPU.

Comparison with State-of-the-Art Methods

Our method specifically targets cross-sensor degradation in real-world pansharpening. Therefore we conduct evaluations in two cross-sensor cases: QB → GF2 and WV3 → WV2, which denotes models pretrained[2] on data from the prior sensor and tested on data from the latter sensor. Quantitative results are reported in Table 2, while visual examples and corresponding HQNR maps are shown in Figure 5.

The results show that most DL-based methods, excluding zero-shot ones, struggle to generalize across sensors. They sometimes even underperform traditional methods, especially in the QB → GF2 case. In contrast, our method significantly improves the cross-sensor generalization of backbone models, setting a new SOTA in cross-sensor scenarios. For instance, HQNR improves by 5.73% for FusionNet and 11.75% for U2Net in the QB → GF2 case, surpassing zero-shot methods which are typically the most generalizable.

Breaking the Runtime Efficiency Bottleneck

As shown in Table 2, zero-shot methods achieve the top-tier fusion quality in cross-sensor cases, establishing themselves as the leading generalization methods aside from our proposed method. However, their efficiency remains a significant bottleneck, often requiring several minutes to optimize for a single test input. In contrast, our method offers a much more favorable efficiency with better fusion quality. As reported in Table 3, it not only outperforms zero-shot methods in fusion quality across most cases, but also reduces processing time to the sub-second level, which is over 100 times faster than zero-shot methods, and makes it substantially more practical for real-world applications.

[2]Traditional and zero-shot methods do not require pretraining and can be directly applied in cross-sensor settings.

Case	Method	HQNR↑	$D_\lambda \downarrow$	$D_s \downarrow$	Duration↓ (s)
QB ↓ GF-2	PsDip	0.7825	0.1323	0.0981	282.59
	ZS-Pan	0.8925	0.0778	**0.0323**	66.74
	$\text{ERFT}_{\text{FusionNet}}$	**0.8970**	**0.0706**	0.0353	**0.11**
	$\text{ERFT}_{\text{U2Net}}$	0.8264	0.0930	0.0888	0.77
WV-3 ↓ WV-2	PsDip	0.8980	0.0385	0.0659	276.18
	ZS-Pan	0.9112	0.0476	0.0435	67.50
	$\text{ERFT}_{\text{FusionNet}}$	0.9100	**0.0445**	0.0475	**0.13**
	$\text{ERFT}_{\text{U2Net}}$	**0.9128**	0.0617	**0.0274**	0.89

Table 3: Comparison of efficiency and fusion quality in two cross-sensor cases between our method and zero-shot methods that typically yield top performance. All values are averaged over test inputs. (**Bold**: best; Underline: second best)

Ablation Study

We conduct a series of ablation experiments to evaluate the contribution of two key components in our method: the feature tailoring (FT) for improving fusion quality and patch-wise (PW) strategy for boosting generalization efficiency.

(I) Feature Tailoring (FT) The FT module is removed from the ERFT-enhanced network to validate its importance in improving the generalization quality. As shown in Table 4, both FusionNet and U2Net experience substantial HQNR drops (5.75% and 11.76%) after removing the FT module.

(II) Patch-wise Strategy (PW) We also evaluate the impact of the PW strategy by disabling it and allowing the ERFT-enhanced network to train and infer on entire test in-

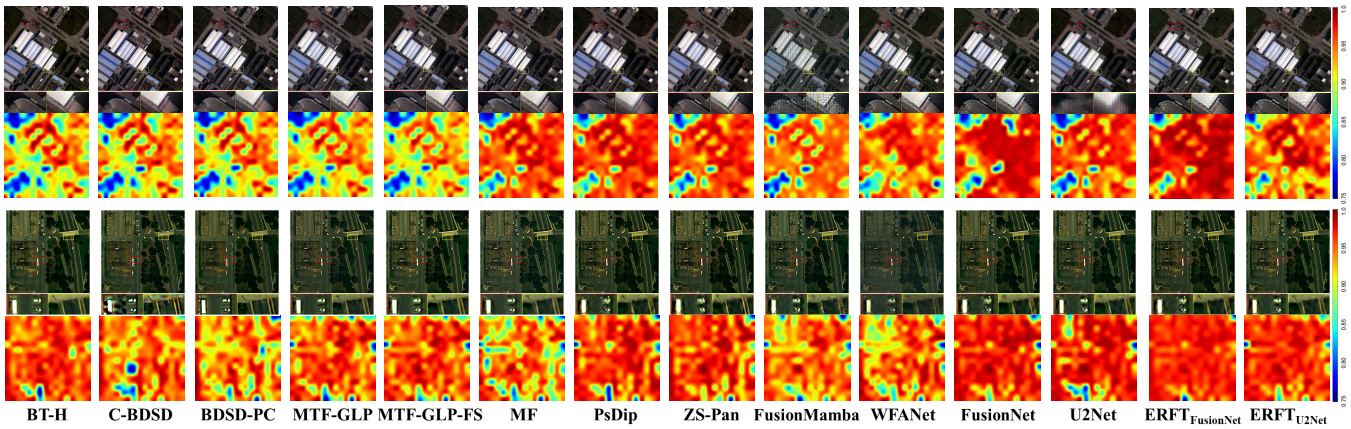

Figure 5: Visual Fusion Examples and HQNR Map in two cross-sensor cases: QB → GF2 (upper) and WV3 → WV2 (lower).

Backbone	FT	PW	HQNR↑	$D_\lambda \downarrow$	$D_s \downarrow$	Duration (s)↓
FusionNet			0.8397	0.1114	0.0551	0.15
	✓		0.9002	0.0698	0.0322	0.45
		✓	0.8395	0.1112	0.0557	0.04
	✓	✓	0.8970	0.0706	0.0353	0.13
U2Net			0.7089	0.2209	0.0909	0.60
	✓		0.8308	0.0873	0.0897	7.53
		✓	0.7088	0.2211	0.0907	0.38
	✓	✓	0.8264	0.0930	0.0888	0.89

Table 4: Ablation study on the effect of Feature Tailor (FT) and Patch-wise strategy (PW) under WV3 → WV2 case.

puts. As reported in Table 4, this setting significantly prolongs processing time, rising to 7.53 seconds for U2Net, thereby failing to achieve the desired sub-second efficiency.

(III) Overall Finally, we compare the ERFT-enhanced networks with their baseline counterparts under both in-sensor and cross-sensor scenarios to evaluate their overall generalization performance. As illustrated in Figure 6, the translucent portions of each bar indicate HQNR gains attributed to ERFT-enhancement. The results demonstrate consistent improvements in fusion quality across different scenarios. Moreover, the near-equal HQNR values between in-sensor and cross-sensor scenarios further highlight the strong generalization capability of our proposed method.

Discussion and Extensive Application

Apart from performance comparison and ablation, we also discuss the influence of hyperparameters including patch size, patch number, training epochs and the method's robostness on newer backbones like FusionMamba and WFANet. Moreover, we extend our method to the novel task of megapixel pansharpening, which processes input images exceeding one million pixels, where our method also achieves SOTA quality with second-level efficiency. Experimental settings and results are available in the *extended version*.

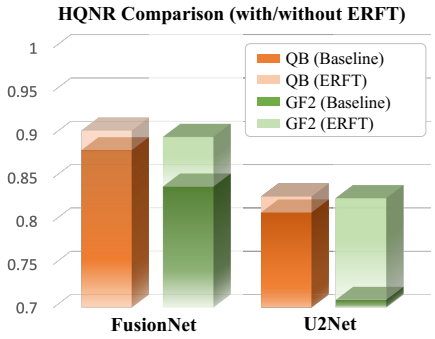

Figure 6: HQNR comparison of FusionNet and U2Net (both pretrained on QB) with and without ERFT enhancement on QB (in-sensor) and GF2 (cross-sensor). The translucent portions indicate the performance gain contributed by ERFT.

Conclusions

This paper presents a novel approach to address cross-sensor degradation in real-world pansharpening: Efficient Residual Feature Tailoring (ERFT). Without relying on extra training data, ERFT achieves superior cross-sensor generalization within sub-second runtime, while fully leveraging pretrained backbones. The method begins by identifying a critical interface via modular decomposition, at which the Feature Tailor module is inserted to learn residual corrections on the fused features, enabling efficient feature-level generalization to test inputs. To further enhance efficiency, ERFT employs a patch-wise strategy during both training and inference, allowing batched parallel processing. This design not only accelerates computation significantly but is also theoretically justified. Extensive experiments demonstrate that ERFT consistently outperforms existing methods in cross-sensor scenarios, achieving SOTA quality with over 100× speedup compared to leading zero-shot methods. Moreover, ERFT scales well to megapixel images while maintaining both quality and efficiency. These results highlight ERFT's strong potential for practical deployment, offering a powerful solution for real-world cross-sensor pansharpening.

Acknowledgments

This research is supported by the Project of the Department of Science and Technology of Sichuan Province (Grant No. 2025YFNH0001).

References

Aiazzi, B.; Alparone, L.; Baronti, S.; Carla, R.; Garzelli, A.; and Santurri, L. 2014. Full-scale assessment of pansharpening methods and data products. In *Image and Signal Processing for Remote Sensing XX*, volume 9244, 924402.

Aiazzi, B.; Alparone, L.; Baronti, S.; Garzelli, A.; and Selva, M. 2006. MTF-tailored multiscale fusion of high-resolution MS and Pan imagery. *Photogrammetric Engineering & Remote Sensing*, 72(5): 591–596.

Cao, Q.; Deng, L.-J.; Wang, W.; Hou, J.; and Vivone, G. 2024a. Zero-shot semi-supervised learning for pansharpening. *Information Fusion*, 101: 102001.

Cao, X.; Fu, X.; Hong, D.; Xu, Z.; and Meng, D. 2021. PanCSC-Net: A model-driven deep unfolding method for pansharpening. *IEEE Transactions on Geoscience and Remote Sensing*, 60: 1–13.

Cao, Z.; Cao, S.; Deng, L.-J.; Wu, X.; Hou, J.; and Vivone, G. 2024b. Diffusion model with disentangled modulations for sharpening multispectral and hyperspectral images. *Information Fusion*, 104: 102158.

Cao, Z.; Zhong, Y.; Wang, Z.; and Deng, L.-J. 2025a. MMAIF: Multi-task and Multi-degradation All-in-One for Image Fusion with Language Guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 11744–11754.

Cao, Z.-H.; Liang, Y.-J.; Deng, L.-J.; and Vivone, G. 2025b. An Efficient Image Fusion Network Exploiting Unifying Language and Mask Guidance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Choi, J.; Yu, K.; and Kim, Y. 2011. A New Adaptive Component-Substitution-Based Satellite Image Fusion by Using Partial Replacement. *IEEE Transactions on Geoscience and Remote Sensing*, 49(1): 295–309.

Deng, L.-J.; Vivone, G.; Jin, C.; and Chanussot, J. 2020. Detail injection-based deep convolutional neural networks for pansharpening. *IEEE Transactions on Geoscience and Remote Sensing*, 59(8): 6995–7010.

Dou, Q.; Castro, D. C.; Kamnitsas, K.; and Glocker, B. 2019. *Domain generalization via model-agnostic learning of semantic features*.

Fu, X.; Lin, Z.; Huang, Y.; and Ding, X. 2019. A Variational Pan-Sharpening With Local Gradient Constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10257–10266. IEEE.

Garzelli, A. 2014. Pansharpening of multispectral images based on nonlocal parameter optimization. *IEEE Transactions on Geoscience and Remote Sensing*, 53(4): 2096–2107.

He, L.; Rao, Y.; Li, J.; Chanussot, J.; Plaza, A.; Zhu, J.; and Li, B. 2019. Pansharpening via detail injection based convolutional neural networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(4): 1188–1204.

Huang, J.; Huang, R.; Xu, J.; Pen, S.; Duan, Y.; and Deng, L. 2025. Wavelet-Assisted Multi-Frequency Attention Network for Pansharpening. arXiv:2502.04903.

Ilyas, A.; Santurkar, S.; Tsipras, D.; Engstrom, L.; Tran, B.; and Madry, A. 2019. *Adversarial examples are not bugs, they are features*. Curran Associates Inc.

Li, W.; Ma, Z.; Deng, L.-J.; Fan, X.; and Tian, Y. 2023. Neuron-Based Spiking Transmission and Reasoning Network for Robust Image-Text Retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(7): 3516–3528.

Li, Z.; Chen, H.; Li, J.; et al. 2024. FusFormer: global and detail feature fusion transformer for semantic segmentation of small objects. *Multimedia Tools and Applications*, 83: 88717–88744.

Liang, Y.; Zhang, P.; Mei, Y.; and Wang, T. 2022. PMAC-Net: Parallel Multiscale Attention Constraint Network for Pan-Sharpening. *IEEE Geoscience and Remote Sensing Letters*, 19: 1–5.

Liu, J.; Li, X.; Wang, Z.; Jiang, Z.; Zhong, W.; Fan, W.; and Xu, B. 2024. PromptFusion: Harmonized semantic prompt learning for infrared and visible image fusion. *IEEE/CAA Journal of Automatica Sinica*.

Lolli, S.; Alparone, L.; Garzelli, A.; and Vivone, G. 2017. Haze correction for contrast-based multispectral pansharpening. *IEEE Geoscience and Remote Sensing Letters*, 14(12): 2255–2259.

Meng, Q.; Shi, W.; Li, S.; and Zhang, L. 2023. PanDiff: A novel pansharpening method based on denoising diffusion probabilistic model. *IEEE Transactions on Geoscience and Remote Sensing*.

Peng, S.; Guo, C.; Wu, X.; and Deng, L.-J. 2023. U2Net: A General Framework with Spatial-Spectral-Integrated Double U-Net for Image Fusion. In *Proceedings of the 31st ACM International Conference on Multimedia*, 3219–3227. Association for Computing Machinery. ISBN 9798400701085.

Peng, S.; Zhu, X.; Deng, H.; Deng, L.-J.; and Lei, Z. 2024. FusionMamba: Efficient Remote Sensing Image Fusion With State Space Model. *IEEE Transactions on Geoscience and Remote Sensing*, 62: 1–16.

Restaino, R.; Vivone, G.; Dalla Mura, M.; and Chanussot, J. 2016. Fusion of multispectral and panchromatic images based on morphological operators. *IEEE Transactions on Image Processing*, 25(6): 2882–2895.

Rui, X.; Cao, X.; Li, Y.; and Meng, D. 2024. Variational Zero-Shot Multispectral Pansharpening. *IEEE Transactions on Geoscience and Remote Sensing*, 62: 1–16.

Tian, X.; Chen, Y.; Yang, C.; and Ma, J. 2022. Variational Pansharpening by Exploiting Cartoon-Texture Similarities. *IEEE Transactions on Geoscience and Remote Sensing*, 60: 1–16.

Vivone, G. 2019a. Robust Band-Dependent Spatial-Detail Approaches for Panchromatic Sharpening. *IEEE Transactions on Geoscience and Remote Sensing*, 6421–6433.

Vivone, G. 2019b. Robust band-dependent spatial-detail approaches for panchromatic sharpening. *IEEE transactions on Geoscience and Remote Sensing*, 57(9): 6421–6433.

Vivone, G.; Restaino, R.; and Chanussot, J. 2018a. Full Scale Regression-Based Injection Coefficients for Panchromatic Sharpening. *IEEE Transactions on Image Processing*, 27(7): 3418–3431.

Vivone, G.; Restaino, R.; and Chanussot, J. 2018b. Full scale regression-based injection coefficients for panchromatic sharpening. *IEEE Transactions on Image Processing*, 27(7): 3418–3431.

Vivone, G.; Restaino, R.; Dalla Mura, M.; Licciardi, G.; and Chanussot, J. 2014. Contrast and Error-Based Fusion Schemes for Multispectral Image Pansharpening. *IEEE Geoscience and Remote Sensing Letters*, 11(5): 930–934.

von Kügelgen, J.; Ketterer, J.; Shen, X.; Meinshausen, N.; and Peters, J. 2025. Representation Learning for Distributional Perturbation Extrapolation. arXiv:2504.18522.

Wang, H.; Gong, M.; Mei, X.; Zhang, H.; and Ma, J. 2024. Deep unfolded network with intrinsic supervision for pansharpening. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 5419–5426.

Wu, Z.-C.; Huang, T.-Z.; Deng, L.-J.; Hu, J.-F.; and Vivone, G. 2022. VO+Net: An Adaptive Approach Using Variational Optimization and Deep Learning for Panchromatic Sharpening. *IEEE Transactions on Geoscience and Remote Sensing*, 60: 1–16.

Xiao, J.-L.; Huang, T.-Z.; Deng, L.-J.; Wu, Z.-C.; and Vivone, G. 2022. A New Context-Aware Details Injection Fidelity With Adaptive Coefficients Estimation for Variational Pansharpening. *IEEE Transactions on Geoscience and Remote Sensing*, 60: 1–15.

Xiao, J.-L.; Huang, T.-Z.; Deng, L.-J.; Wu, Z.-C.; Wu, X.; and Vivone, G. 2023. Variational pansharpening based on coefficient estimation with nonlocal regression. *IEEE Transactions on Geoscience and Remote Sensing*, 61: 1–15.

Yang, J.; Fu, X.; Hu, Y.; Huang, Y.; Ding, X.; and Paisley, J. 2017. PanNet: A Deep Network Architecture for Pan-Sharpening. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1753–1761.

Yosinski, J.; Clune, J.; Bengio, Y.; and Lipson, H. 2014. How transferable are features in deep neural networks? In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, 3320–3328.

Zhang, H.; and Ma, J. 2021. GTP-PNet: A residual learning network based on gradient transformation prior for pansharpening. *ISPRS Journal of Photogrammetry and Remote Sensing*, 172: 223–239.

Zhang, Y.; Meng, Q.; Shi, W.; Li, S.; and Zhang, L. 2024. SSDiff: A Spatial-Spectral Integrated Diffusion Model for Remote Sensing Pansharpening. In *Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS)*.

Zhong, Y.; Wu, X.; Deng, L.-J.; Cao, Z.; and Dou, H.-X. 2024. Ssdiff: Spatial-spectral integrated diffusion model for remote sensing pansharpening. *Advances in Neural Information Processing Systems*, 37: 77962–77986.

Zhou, H.; Liu, Q.; and Wang, Y. 2022. Panformer: A transformer based model for pan-sharpening. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, 1–6. IEEE.