SWIFT: A General Sensitive Weight Identification Framework for Fast Sensor-Transfer Pansharpening

Zeyu Xia[1*], Chenxi Sun[1*], Tianyu Xin[1*], Yubo Zeng[1], Haoyu Chen[1], Liang-Jian Deng[1†]

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

Abstract

Although deep learning-based methods have achieved promising performance in Pansharpening, they generally suffer from severe performance degradation when applied to data from unseen sensors. Existing cross-domain strategies, including retraining, fine-tuning, and zero-shot methods, fail to simultaneously preserve model architecture and maintain low adaptation costs. Therefore, we are the first to define and address a novel task in the pansharpening field: enhancing a model's cross-sensor generalization at an extremely low cost while keeping the model architecture invariant. To tackle this task, we propose SWIFT (Sensitive Weight Identification for Fast Transfer), a plug-and-play framework. SWIFT first employs an unsupervised manifold-based sampling strategy to efficiently select a high-fidelity subset the most informative target-domain samples. It then leverages this subset to probe a source-domain pre-trained model, identifying and updating only the weight subset most sensitive to the domain shift by analyzing the gradient behavior of its parameters. Extensive experiments demonstrate that SWIFT can be applied to various deep learning models, boosting adaptation efficiency by up to 30-fold. On a single NVIDIA RTX 4090 GPU, this reduces adaptation time from hours to as little as one minute. The adapted models not only substantially outperform direct-transfer baselines but also achieve performance competitive with, or even superior to full retraining while using only 3% of the target domain dataset and adapting nearly 10% to 30% of the model's parameters. This establishes a new state-of-the-art on the WorldView-2 and QuickBird datasets.

Code Availability:https://github.com/Eden-netizen/SWIFT

Introduction

High-resolution multispectral (HRMS) images are essential for applications such as urban planning and environmental monitoring (Crampton et al. 2013; Fitzner et al. 2013). However, due to hardware limitations, many commercial satellites such as WorldView-2 (WV2) and Gaofen-2 (GF2) cannot capture images with both high spatial resolution and rich spectral information. To resolve this trade-off, these systems acquire two complementary data types: high-resolution panchromatic (PAN) and low-resolution multispectral (LRMS) images. The process of fusing these two kinds of images to generate high-resolution multispectral (HRMS) images is known as Pansharpening.

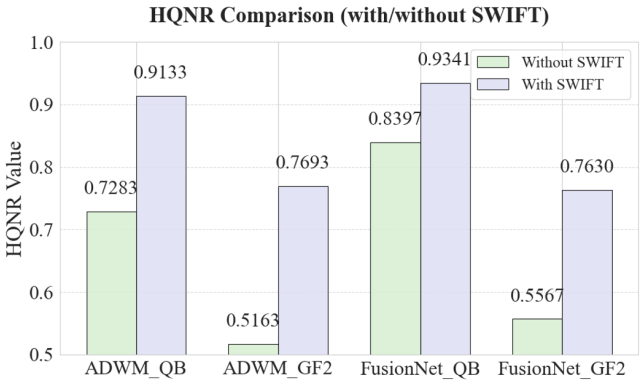

Figure 1: The SWIFT framework and its impact on performance. Top: The two-step strategy of SWIFT. Bottom: Comparison of HQNR between two representative models (i.e., FusionNet (Deng et al. 2021) and ADWM (Huang et al. 2025a)) with and without SWIFT enhancement.

Over the past few decades, Pansharpening methods have evolved considerably, transitioning from traditional approaches to modern deep learning-based methods. Traditional approaches can be roughly categorized into three primary classes: component substitution (CS)(Choi et al. 2005; Vivone et al. 2015), multiresolution analysis (MRA)(Otazu et al. 2005; Vivone, Restaino, and Chanussot 2018), and variational optimization (VO)(Tian et al. 2022; Wu et al. 2025). More recently, deep learning techniques have leveraged the powerful feature extraction capabilities of neural architectures. Models based on Convolutional Neural Networks (CNN) (Masi et al. 2016; He, Zhong, and Ma 2019; Yang et al. 2023), Transformer (Zhang and Ma 2021; Zhou, Liu, and Wang 2022; Li et al. 2024b; Wu et al. 2025), and diffusion (Meng et al. 2023; Zhong et al. 2024; Rui et al. 2024) have already produced superior fusion results, significantly outperforming traditional methods.

While deep learning (DL) methods(Li et al. 2024a, 2025) have become the dominant paradigm in Pansharpening, they suffer from a critical limitation: severe performance degra-

*These authors contributed equally.
†Corresponding author.

dation when applied to target-domain data that exhibits a distribution shift from the source. This "cross-domain generalization" challenge is exacerbated by the diversification of satellite sensors and has led to a fundamental performance-cost dilemma. Current solutions either involve designing more complex architectures, which increases computational load without guaranteeing better generalization, as our experiments with models like FusionNet and ADWM demonstrate, and full-scale retraining on the target domain is also expensive. Consequently, a critical research gap exists: no current method effectively balances adaptation cost with architectural invariance.

Based on the preceding analysis, this paper introduces a novel task in Pansharpening field: improving a model's generalization at a low computational cost while maintaining its original architecture. We propose a *model-agnostic enhancement framework*—SWIFT to firstly address this task. It first employs a density-aware sampling method to select a minimal, high-information data subset. Subsequently, SWIFT leverages this data to perform a gradient-based sensitivity analysis, identifying and updating only the most sensitive model parameters, thereby achieving efficient and precise model adaptation, as we show in Figure 1.

In summary, the contributions of this study are as follows:

- We are the first to define and address a novel task in the Pansharpening field that enhances model generalization while jointly considering model architecture invariance and adaptation economy, offering a more practical research perspective for the community.
- We propose SWIFT, a plug-and-play, general-purpose framework. It integrates: (a) a data density-aware sample selection strategy, and (b) an efficient, gradient-based parameter sensitivity identification method, providing the first effective solution to the newly defined task.
- Extensive experiments demonstrate that SWIFT significantly improves cross-sensor performance of various models, achieving results *comparable to or even surpassing* full retraining on the target domain while requiring only minimal adaptation cost (nearly one minute, 3% training data, and 10% to 30% trainable parameters), indicating the effectiveness of our framework.

Related Work

Deep Learning-based Pansharpening

Deep learning is the dominant paradigm in Pansharpening field. Based on network architecture, these methods can be broadly classified into three categories. Early CNN-based models, from the pioneering PNN (Masi et al. 2016) to advanced architectures like PanNet (Yang et al. 2017) and FusionNet (Deng et al. 2021), excel at capturing local spatial-spectral patterns, but their limited receptive fields restrict the modeling of global dependencies. Transformer-based methods such as PanFormer (Zhou, Liu, and Wang 2022) incorporating self-attention to model long-range dependencies. Although this improves fusion accuracy, the high computational cost limits their applicability in resource-constrained scenarios. Finally, emerging hybrid methods, such as diffusion-based SSDiff (Zhong et al. 2024) that have emerged to capture complex data distributions, but these models are usually sensor-specific and require costly retraining for adaptation. Consequently, a core bottleneck in all deep learning models persists: a model trained in a source domain (e.g., QuickBird) often suffers from spectral distortion or spatial detail loss when applied to a target domain (e.g., GaoFen-2) due to distribution shifts caused by different sensor characteristics.

Existing Cross-Domain Strategies

To address the performance degradatio in cross-domain scenarios, existing research follows four technical routes:

Model retraining and architectural optimization improve performance through full retraining (e.g., PanNet (Yang et al. 2017) retrained on target domain data) or architectural adjustments (e.g., FusionNet (Deng et al. 2021) adding detail modules, ADWM (Huang et al. 2025a) introducing dual-level mechanisms). However, they require large amounts of data and lengthy training, while architectural modifications will hinder lightweight and weight-only updates for deployed models. Parameter-Efficient Fine-Tuning (PEFT) freezes the backbone network and updates few parameters. For example, PanAdapter (Wu et al. 2025) in the pansharpening domain inserts lightweight adapter modules, and LoRA (Hu et al. 2021) in the general domain uses low-rank matrices to adjust key layers. Although validated effective in vision models, adapter methods like PanAdapter require modifications to the model architecture, LoRA relies on empirical selection of low-rank dimensions, and it remains challenging to capture target domain distributions accurately with *extremely few samples*, all of which limit adaptation efficiency.

Zero-shot methods (e.g., ZS-Pan (Cao et al. 2024; Zi-Han Cao 2025) in Pansharpening) perform inference optimization using only a single target-domain image pair without additional labels. Yet, single-image processing is time-consuming and the fusion quality is limited, failing to meet demands for timeliness and stability. Data augmentation and selection enhance adaptability by expanding diversity or selecting representative samples, but existing strategies based on simple distance metrics or uniform distribution assumptions struggle to balance sparse edge and dense core features in minimal sample sets, leading to insufficient learning of target domains.

Methodology

Motivation

The preceding analysis of related work reveals a fundamental performance-cost dilemma in cross-domain scenarios. Full retraining is prohibitively expensive, while architectural modifications hinder lightweight, weight-only updates for deployed models. Moreover, zero-shot and PEFT methods are often inefficient and yield suboptimal fusion quality. Consequently, two key challenges remain: reducing adaptation costs while preserving the original model architecture. To our knowledge, no existing method effectively addresses these two mutually constraining challenges simultaneously.

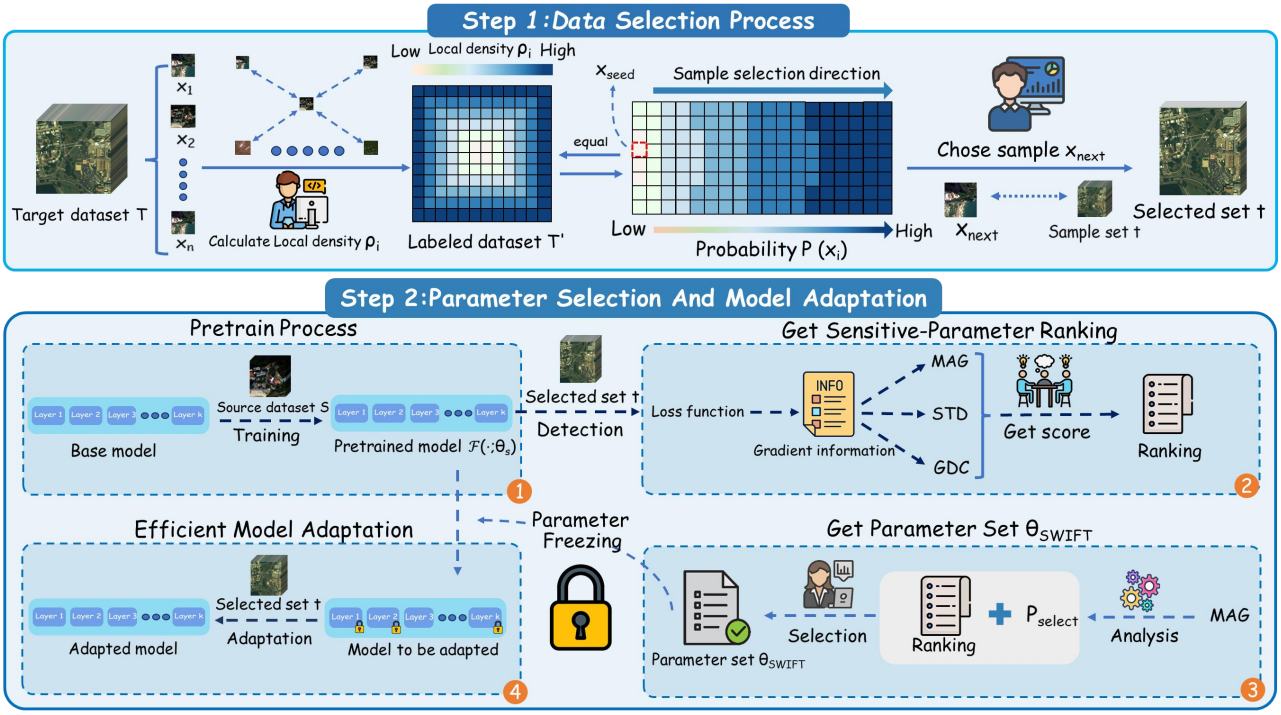

Figure 2: Overall Framework of the SWIFT framework. Step 1 first selects low-density samples and then chooses high-distance samples via Farthest Point Sampling (FPS) to obtain a high-information subset t. Step 2 then selects the most sensitive parameters by analyzing gradient information and uses the subset t to adapt the pre-trained model.

Motivated by this gap, we are the first to define and address a novel task in the Pansharpening field: enhancing a model's cross-sensor generalization at a minimal cost while maintaining architectural invariance. To tackle this task, we propose SWIFT, a framework built on a different philosophy. We posit that the key to efficient adaptation lies not only in which parameters to update, but also in which data to use for the update, thus introducing *data efficiency* as a new, crucial dimension largely ignored by conventional PEFT. SWIFT operationalizes this insight through a two-step "targeted identification" strategy that reduces computational costs without altering the model architecture. The design of SWIFT is introduced in the following subsection and illustrated in Figure 2. A complete list of all hyperparameters and implementation details can be found in the supplementary material.

Density-Aware Farthest Point Sampling Strategy

In cross-domain model adaptation, the informational value of target-domain samples is heterogeneous: samples in dense regions are highly redundant, while critical information in sparse regions is easily overlooked. Thus, we proposed the Density-Aware Farthest Point Sampling (DA-FPS) strategy to get a small subset that preserves the core distribution of the target domain. Figure 3 shows the effectiveness.

First, to quantify the "uniqueness" of each sample, we construct a unified feature vector x_i by concatenating the flattened vectors of its corresponding PAN, upsampled

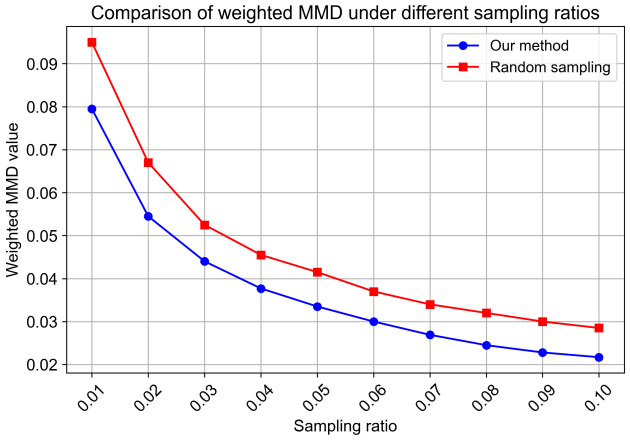

Figure 3: Comparison of the Maximum Mean Discrepancy (MMD) (Gretton et al. 2012) between our sampling method and random sampling under different sampling ratios.

LRMS, and original MS images, then calculate the local density ρ_i for each sample x_i in the target-domain dataset $T = \{x_1, x_2, \ldots, x_N\}$ by using a K-Nearest Neighbors (kNN) based approach:

$$\rho_i = \frac{k}{\sum_{x_j \in N_k(x_i)} d(x_i, x_j) + \epsilon} \quad (1)$$

where $d(x_i, x_j)$ denotes the Euclidean distance between samples x_i and x_j:

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{D}(x_{i,k} - x_{j,k})^2} \tag{2}$$

where $N_k(x_i)$ is the set of the k nearest neighbors of sample x_i, and ϵ is a small constant to prevent division by zero.

In our experiments, we set k = 5. The impact of this hyperparameter is further analyzed in the supplementary material. A larger ρ_i indicates that the sample x_i is located in a dense region, while a smaller ρ_i suggests that it is in a sparse and more informative region. Subsequently, we obtain get a dataset labeled by local density $T' = \{x_1, x_2, \ldots, x_N\}$.

After obtaining the density information for all samples, we iteratively construct the "essence" sample set t. Our selection process is conceptually guided by a balance between data uniqueness and spatial coverage. Specifically, we leverage the calculated density ρ_i to guide the selection of a high potential candidate set, denoted as $C \subset T'$,the probability $P(x_i)$ of each sample x being selected into the candidate set is defined as:

$$P(x_i) = \frac{1/\rho_i}{\sum_{j=1}^{N}(1/\rho_j)} \tag{3}$$

which ensures that samples from sparse regions are more likely to be considered.

Finally, we apply an efficient Farthest Point Sampling (FPS) algorithm on this candidate set C, this process is initialized by selecting the sample with the lowest global density to be the initial seed, which ensure that the rarest information is not missed. In each subsequent iteration, we select the next sample x_{next} from every unselected sample $x_i \in C \setminus t$, the selection criterion is formally expressed as:

$$x_{next} = \arg\max_{x_i \in C \setminus t} \left(\min_{s \in t} \|x_i - s\|_2 \right) \tag{4}$$

The chosen sample is the one that maximizes the shortest distance to any sample already in t. This process continues until t reaches a predefined size. The resulting subset t thus preserves the core distribution of the target domain, benefiting from both the density-based pre-selection and the distance-based coverage maximization.

Sensitive Parameter Selection and Efficient Model Adaptation

After obtaining the high-information sample subset t, we further screen the parameters in the pretrained model $\mathcal{F}(\cdot\,; \theta_S)$ that are most sensitive to target domain shifts, and only update this subset to achieve efficient adaptation.

As we all know, in the field of deep learning, the rule for parameter updates is as follows

$$\theta_j \leftarrow \theta_j - \eta \nabla_{\theta_j} \mathcal{L} \tag{5}$$

where the magnitude of gradient determines the convergence rate of the loss function directly, this principle is widely recognized and utilized in pruning work. However, the magnitude or direction of the gradient for a single parameter can vary significantly across different data batches, so we introduced Standard Deviation(STD) and Direction Consistency(GDC) to measure the stability of the gradient, which allow us to select parameters that are both impactful and consistently stable. The introduction of the above two variables can also relatively mitigate the influence of gradient explosion on parameter selection.

Specifically, to quantify these characteristics, we divide the "essence" sample subset t into M microbatches. For any given trainable parameter θ_j in the model, we perform a full forward and backward pass for each microbatch i to obtain a set of gradients $G_j = \{g_{j,1}.g_{j,2}, \ldots, g_{j,M}\}$. Based on this set, we compute the following three core metrics:

First, calculate the Gradient Magnitude (MAG), which reflects the parameter importance through the average intensity of gradients. A larger magnitude indicates a higher impact on the loss reduction. It is formulated as follows:

$$\mathrm{MAG}(\theta_j) = \frac{1}{M} \sum_{i=1}^{M} |E[g_{j,i}]| \tag{6}$$

where $E[g_{j,i}]$ is the expectation of all gradient values for parameter θ_j within the i-th microbatch.

Second, we compute the *Gradient Direction Consistency (GDC)*, which is used to evaluate the stability of parameter update directions, and is defined as:

$$\mathrm{GDC}(\theta_j) = \frac{1}{M} \sum_{i=1}^{M} \frac{\max(N_{pos}, N_{neg})}{N_{pos} + N_{neg}} \tag{7}$$

where N_{pos} and N_{neg} are the counts of positive and negative gradients for parameter θ_j in the i-th microbatch, respectively. This metric measures the consistency of the expected update direction across different samples. A GDC value closer to 1 indicates a more definite update direction.

Finally, we calculate the *Gradient Standard Deviation (STD)*, which is used to characterize the intensity of gradient fluctuations, with the formula::

$$\mathrm{STD}(\theta_j) = \sqrt{\frac{1}{M} \sum_{i=1}^{M} \mathrm{Var}(g_{j,i})} \tag{8}$$

where $Var(g_{j,i})$ is the variance of the gradients for parameter θ_j in the i-th microbatch.This metric reflects the stability of the update intention; a lower STD signifies a more stable update process.

To synthesize the importance of these three metrics, we normalize them and devise a weighted sum to compute a composite sensitivity score for each trainable parameter θ_j:

$$S(\theta_j) = \alpha \cdot \overline{\mathrm{MAG}(\theta_j)} + \beta \cdot (1 - \overline{\mathrm{STD}(\theta_j)}) + \gamma \cdot \overline{\mathrm{GDC}(\theta_j)} \tag{9}$$

where $\overline{\mathrm{MAG}}$, $\overline{\mathrm{STD}}$, and $\overline{\mathrm{GDC}}$ are the normalized metrics, we set these weights to α = 0.6, β = 0.1, γ = 0.3, respectively, a detailed discussion on the selection of these values is provided in our experiment part. Notably, we assign a negative weight to the standard deviation term to reward parameters with more stable gradients.

After obtaining and ranking the sensitivity scores for all parameters in descending order, we employ a dynamic thresholding mechanism for parameter selection. First, quantify the "sharpness" of the gradient distribution to judge parameter distinguishability, Let $\mathcal{M} = \{\overline{\mathrm{MAG}_j}\}$ denote the set of normalized gradients:

$$\mathcal{H} = \mathrm{std}(\mathcal{M}) + (\max(\mathcal{M}) - \mathrm{median}(\mathcal{M})) \tag{10}$$

where $\{\overline{\mathrm{MAG}_j}\}$ is the set of normalized average gradient magnitudes across all parameters. The sharpness is composed of two complementary components: the standard deviation and the difference between the maximum and median, a high *std* means that the distribution of gradient magnitudes is very uneven, but it is insufficient as it does not capture the *shape* of the distribution. For instance, a wide, symmetric distribution and a skewed distribution with a long tail could have similar *std* values. Therefore, we introduce the *difference* mentioned above, a large difference specifically identifies a "sharp head" in the distribution, complementing the *std* by capturing the upper-tail skewness. A higher $\mathcal{H}$ value indicates a clear distinction between important and unimportant parameters. Subsequently, based on the calculated sharpness $\mathcal{H}$, we dynamically determine the optimal selection ratio P_{select} for the current adaptation task to identify the final parameter subset θ_{SWIFT}:

$$\mathbf{P}_{select} = \eta_{\min} + (\eta_{\max} - \eta_{\min}) \cdot \mathrm{clip}(\mathcal{H}_{norm}, 0, 1) \tag{11}$$

where η_{min} and η_{max} are preset ratio ranges(set to 0 and 1 respectively in our experiments), and $\mathcal{H}_{norm}$ is the min-max normalized result of $\mathcal{H}$, defined as:

$$\mathcal{H}_{norm} = \frac{\mathcal{H} - \mathcal{H}_{\min}}{\mathcal{H}_{\max} - \mathcal{H}_{\min}} \tag{12}$$

Finally, we freeze all trainable parameters outside the subset θ_{SWIFT} and update this sensitive subset using the sample subset t to achieve efficient model adaptation.

Experiment

Datasets and Metrics

Datasets: We investigate the effectiveness of the proposed method on a wide range of datasets, including an 8 band dataset from WorldView-3 (WV3) and WorldView-2 (WV2) sensors, and 4-band datasets from QuickBird (QB) and GaoFen-2 sensors. Notably, we leverage Wald's protocol to simulate the source data due to the unavailability of ground truth (GT) images. Taking WV3 as an instance, we use 10000 PAN/LRM S/GT image pairs (64 $\times$ 64) for network training. For testing, we take 20 PAN/LRMS/GT image pairs (256 $\times$ 256) for reduced-resolution evaluation, and 20 PAN/LRMS image pairs (512 $\times$ 512) for the full-resolution assessment, which lacks GT images.

Metrics: The quality evaluation is conducted at two resolutions. For reduced resolution tests, the widely used SAM (Yuhas, Goetz, and Boardman 1992), ERGAS (Wald 2002), SCC (Zhou, Civco, and Silander 1998), and Q-index for 4-band (Q4) and 8-band data (Q8) (Garzelli and Nencini 2009) are adopted to assess the quality of the results. To evaluate the performance at full resolution, the HQNR, the Dλ, and the Ds (Vivone et al. 2015) indexes are considered.

Training Details and Benchmark

Training Details: We denote our experimental settings as "SourceDomain→TargetDomain". In our result tables, *Model* indicates it is trained from scratch and tested on TargetDomain, $Model_{cross}$ or $SWIFT_{Model}$ donates it is pretrained on the SourceDomain, then transfer to and finally tested on TargetDomain. All experiments are conducted on a NVIDIA RTX 4090 GPU with 24GB of video memory.

Benchmark: We compare our method with a series of representative deep learning models in Pansharpening field, covering architectures from classical to the latest ones, including PanNet (Yang et al. 2017), FusionNet (Deng et al. 2021), U2Net (Peng et al. 2023), SSDiff (Zhong et al. 2024), ADWM (Huang et al. 2025a), and WFANet (Huang et al. 2025b). For fairness, all baseline models are pretrained by the same source-domain dataset as our method, and their hyperparameter settings strictly follow the configurations in their respective original papers. Notably, we exclude Zero-Shot methods like ZS-Pan and PSDip from this direct comparison as they *do not utilize training data*. Similarly, PEFT methods like Panadapter are not included in our main comparison because their baseline is a general super-resolution model, which differs from our task of adapting existing *Pansharpening-specific* models, but we still make a comparison as they are both key cross-domain methods.

Main Experimental Results

Our main experimental results, presented in Tables 1, 3, 4, demonstrate that the SWIFT framework significantly enhances the cross-sensor performance of various Pansharpening models. While models perform excellently on source-domain data, their performance degrades significantly when applied to a target domain with a distribution shift, as shown in Table 1. A pretrained model shift to a target domain(e.g., $\mathrm{PanNet}_{\mathrm{cross}}$) always has a huge performence degration compared to a model trained on target domain(e.g., PanNet) directly. Based on this, enhancing the performance of source pretrained models has become a key goal for improving cross-domain generalization ability.

Our method establishes a new SOTA performance on the HQNR metric for WV2 and QB datasets. Qualitatively, the visual comparisons in Figure 4 show that SWIFT-enhanced networks consistently produce results with error maps exhibiting visibly smaller residuals. Notably, these results also validate our initial claim from the introduction: as shown in our cross-domain experiments (e.g., Table 1), the FusionNet model consistently outperforms the ADWM which is designed above FusionNet, confirming that increasing architectural complexity does not guarantee better generalization.

Quantitatively, the improvements are substantial across all tested datasets. For instance, when SWIFT is applied to ADWM, the HQNR metric is significantly increased by 0.253 on the full-resolution QB data (Table 1). Similarly, our method has also achieved remarkable results on reduced-resolution data (details in the supplementary materials). **Additionally,** it is crucial to note that *the Time(s) metric signifies different procedures*: for baseline models (e.g., PanNet,

Method	QB → GF2: Avg±std				GF2 → QB: Avg±std			
	$D_\lambda \downarrow$	$D_s \downarrow$	HQNR↑	Time(s)	$D_\lambda \downarrow$	$D_s \downarrow$	HQNR↑	Time(s)
Panadapter	0.018±0.017	0.070±0.012	0.903±0.020	74337.1	0.033±0.018	0.061±0.014	0.837±0.029	71937.2
ZS-Pan	0.078±0.018	0.032±0.016	0.893±0.024	1636.2	0.036±0.016	0.071±0.019	0.896±0.027	1334.8
PSDip	0.132±0.029	0.098±0.026	0.783±0.033	6630.8	0.045±0.021	0.075±0.021	0.883±0.022	5651.8
PanNet	0.123±0.024	0.190±0.019	0.712±0.025	12870.6	0.104±0.026	0.078±0.012	0.826±0.031	15486.4
$\text{PanNet}_{\text{cross}}$	0.220±0.040	0.198±0.026	0.625±0.023	12870.6	0.285±0.045	0.232±0.023	0.549±0.026	15486.4
$\text{SWIFT}_{\text{PanNet}}$	0.090±0.019	0.056±0.012	0.860±0.026	1399.6	0.193±0.059	0.172±0.025	0.668±0.023	1630.4
FusionNet	0.035±0.019	0.103±0.013	0.866±0.022	5874.3	0.067±0.020	0.046±0.014	0.890±0.029	6435.2
$\text{FusionNet}_{\text{cross}}$	0.111±0.050	0.055±0.014	0.840±0.051	5874.3	0.311±0.045	0.191±0.039	0.557±0.033	6435.2
$\text{SWIFT}_{\text{FusionNet}}$	0.041±0.037	**0.027±0.006**	**0.934±0.035**	**130.7**	0.167±0.050	0.048±0.016	0.794±0.056	**138.6**
U2Net	0.020±0.012	0.046±0.010	0.936±0.013	11023.3	0.075±0.018	0.047±0.023	0.882±0.036	12769.2
$\text{U2Net}_{\text{cross}}$	0.137±0.047	0.134±0.038	0.747±0.052	11023.3	0.105±0.057	0.075±0.029	0.828±0.054	12769.2
$\text{SWIFT}_{\text{U2Net}}$	0.033±0.020	0.071±0.013	0.898±0.017	460.2	0.132±0.040	0.041±0.031	0.834±0.059	572.4
SSDiff	0.021±0.011	0.061±0.017	0.920±0.021	32336.5	0.032±0.011	0.037±0.012	0.933±0.020	35972.1
$\text{SSDiff}_{\text{cross}}$	0.144±0.075	0.060±0.041	0.805±0.084	32336.5	0.082±0.038	0.076±0.025	0.850±0.047	35972.1
$\text{SWIFT}_{\text{SSDiff}}$	0.033±0.029	0.036±0.015	0.932±0.021	1077.9	**0.033±0.012**	**0.026±0.016**	**0.942±0.026**	1199.1
ADWM	0.104±0.151	0.079±0.035	0.830±0.155	13645.9	0.076±0.019	0.039±0.013	0.888±0.030	26284.6
$\text{ADWM}_{\text{cross}}$	0.161±0.046	0.133±0.035	0.728±0.060	13645.9	0.795±0.196	0.127±0.182	0.516±0.177	26284.6
$\text{SWIFT}_{\text{ADWM}}$	0.049±0.034	0.040±0.015	0.913±0.032	1091.3	0.107±0.035	0.140±0.030	0.769±0.055	971.7
WFANet	0.017±0.008	0.064±0.010	0.920±0.011	24994.5	0.044±0.019	0.110±0.024	0.851±0.013	26787.0
$\text{WFANet}_{\text{cross}}$	0.035±0.012	0.066±0.019	0.900±0.012	24994.5	0.304±0.082	0.242±0.044	0.526±0.059	26787.0
$\text{SWIFT}_{\text{WFANet}}$	**0.020±0.013**	0.061±0.010	0.920±0.012	1612.4	0.074±0.022	0.085±0.026	0.849±0.042	1856.7

Table 1: Mean values, standard deviations, and time used of the baseline models, cross-domain models, and SWIFT-enhanced models on 20 full-resolution samples from the QB and GF2 datasets. Best results: **bold**, second-best: underline.

$\text{PanNet}_{\text{cross}}$), it is the training or retraining time on the target domain, whereas for our SWIFT-enhanced models (e.g., $\text{SWIFT}_{\text{PanNet}}$), it denotes the rapid adaptation time.This comparison is fair and reflects a practical application scenario, as all methods start from the same pretrained source-domain model; we focus on evaluating the marginal cost required for adaptation: full retraining versus our SWIFT enhancement. And SWIFT requires only 3% of the target-domain samples and adapts nearly 10% to 30% of the model's parameters. More notably, the adaptation time for a model like SSDiff is reduced from nearly 10 hours to just under 20 minutes, an approximately *30-fold* improvement in efficiency (Table 1). These results validates the effectiveness of SWIFT and highlights its comprehensive advantages in balancing performance with data, parameter, and time efficiency.

Ablation Study

To verify the effectiveness of the "essence" sample selection strategy and the key parameter identification and update mechanism, we designed a series of ablation experiments on the QB dataset using the FusionNet baseline model.

Density-Aware Farthest Point Sampling strategy: To verify our DA-FPS strategy, we compared it with random sampling (the 3% ratio, where gradient change stabilizes significantly afterward, was chosen as a key point, and we shows it in Figure 3). Results in Table 2 confirm our method better selects samples with high learning value.

Method	GF2 → QB: Avg±std				
	$D_\lambda \downarrow$	$D_s \downarrow$	HQNR↑	MMD	Time(s)
DA-FPS	0.17±0.05	0.05±0.01	0.79±0.06	0.044	69
Random	0.14±0.04	0.13±0.03	0.75±0.05	0.053	61

Table 2: Comparison of metrics between DA-FPS and random sampling at a 3% ratio on full-resolution samples from the QB dataset using the FusionNet benchmark model.

Method	WV3 → WV2: Avg±std			
	$D_\lambda \downarrow$	$D_s \downarrow$	HQNR↑	Time(s)
FusionNet	0.05±0.03	0.06±0.02	0.89±0.02	2630
$\text{SWIFT}_{\text{FusionNet}}$	**0.03±0.02**	0.04±0.01	0.93±0.01	**78**
U2Net	0.09±0.08	0.04±0.01	0.87±0.08	35129
$\text{SWIFT}_{\text{U2Net}}$	0.04±0.02	**0.02±0.01**	**0.94±0.02**	243
WFANet	0.06±0.04	0.03±0.01	0.91±0.04	16481
$\text{SWIFT}_{\text{WFANet}}$	0.04±0.02	0.03±0.01	0.94±0.02	204

Table 3: Means, standard deviations, and time of all comparative methods on 20 full-resolution samples from the WV2 datasets. Best: **bold**, and second best: underline.

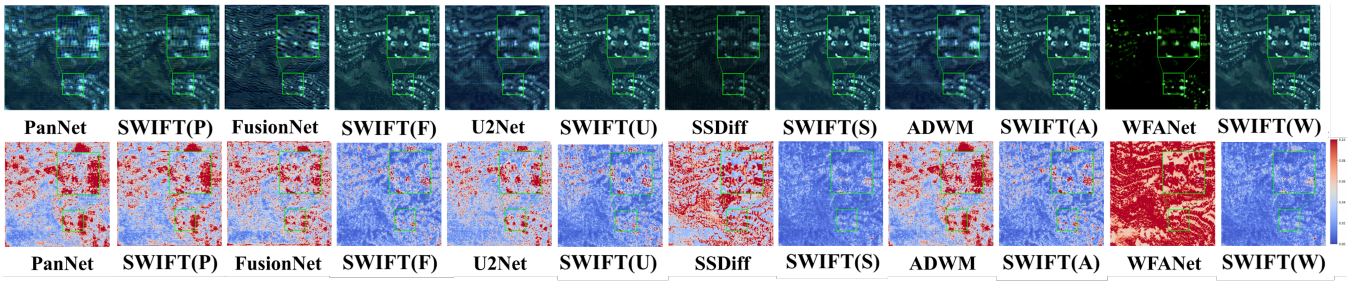

Figure 4: Visual Fusion image and Error maps on QB dataset (reduced data). For the error maps, blue indicate low error.

Dynamic Parameter Selection: To validate our dynamic selection strategy, we compared it against fixed-ratio selection from 10% to 100%. The results in Figure 5 demonstrate that our dynamic method, while retaining only 34.1% of parameters, surpasses the performance of any fixed-ratio selection and even matches that of full fine-tuning, but at a fraction of the computational cost.

Method	WV3 → WV2: Avg±std			
	SAM↓	ERGAS↓	SCC↑	Q8↑
FusionNet	6.33±0.65	5.04±0.46	0.88±0.01	0.86±0.08
$\text{SWIFT}_{\text{FusionNet}}$	4.42±0.46	**2.68±0.39**	**0.97±0.01**	**0.91±0.09**
U2Net	5.26±0.50	4.08±0.38	0.93±0.01	0.85±0.09
$\text{SWIFT}_{\text{U2Net}}$	4.50±0.52	2.81±0.44	0.97±0.01	0.91±0.09
WFANet	5.79±0.56	4.35±0.39	0.92±0.01	0.84±0.08
$\text{SWIFT}_{\text{WFANet}}$	**4.38±0.46**	2.70±0.41	0.97±0.01	0.91±0.08

Table 4: Means, standard deviations, and time of all comparative methods on 20 reduced-resolution samples from the WV2 datasets. Best: **bold**, and second best: underline.

Setting			GF2 → QB: Avg±std		
α	β	γ	$D_\lambda \downarrow$	$D_s \downarrow$	HQNR↑
1	0	0	0.126±0.034	0.165±0.026	0.731±0.048
0	1	0	0.125±0.038	0.162±0.027	0.734±0.053
0	0	1	0.111±0.048	0.148±0.026	0.758±0.059
0.33	0.33	0.33	0.153±0.044	0.135±0.026	0.734±0.055
0.6	0.2	0.2	0.153±0.050	0.078±0.024	0.782±0.055
0.6	0.1	0.3	0.091±0.028	0.129±0.031	0.793±0.049

Table 5: Weight hyperparameter settings for gradient magnitude, gradient direction consistency, and gradient standard deviation in parameter sensitivity analysis.

Sensitivity Score Weights: To analyze the hyperparameter sensitivity of our scoring strategy, we performed an ablation study on the weights (α, β, γ). As presented in Table 5, we designed and compared six different hyperparameter configurations. The results confirm that our proposed setting outperforms simpler approaches, such as relying on a single metric or weighting all three metrics equally.

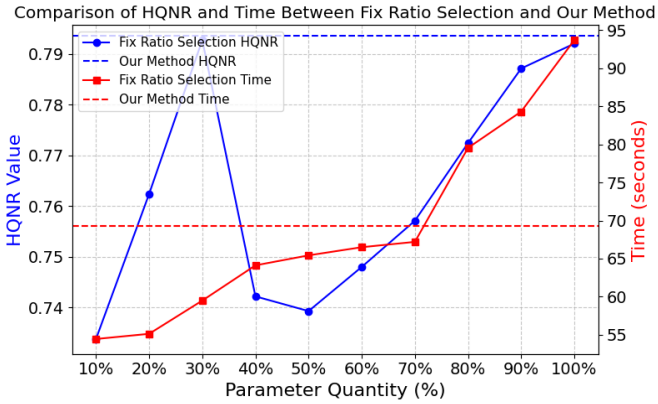

Figure 5: Performance and efficiency comparison of our dynamic parameter selection strategy against fixed-ratio selection. The blue line (left y-axis) shows the HQNR score, while the red line (right y-axis) shows the adaptation time.

Conclusion

This paper introduced SWIFT, a model-agnostic enhancement framework designed to address the novel task of improving cross-sensor generalization at an extremely low cost without altering the model architecture. Extensive experiments demonstrate that SWIFT achieves SOTA performance on the HQNR metric across the WV2 and QB datasets, often surpassing full retraining while using only a fraction of the data (3%) and tunable parameters (10%-30%). By efficiently balancing high performance with minimal adaptation costs, SWIFT offers a new and practical paradigm for deploying Pansharpening models across diverse sensor domains.

Acknowledgments

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

In addition, the author wishes to thank the members of the The **AI Club** and **The Workshop of Innovation and Entrepreneurship** of School of Economy and Menegement of

UESTC for the beneficial discussions; the active academic atmosphere of the club is fertile ground for innovation.

Out of personal sentiment, Zeyu Xia extends his most sincere gratitude to Zixin Wei. Her understanding and patience provided the essential foundation for the successful completion of this work.

References

Cao, Q.; Deng, L.-J.; Wang, W.; Hou, J.; and Vivone, G. 2024. Zero-shot Semi-supervised Learning for Pansharpening. *Information Fusion*.

Choi, M.; Kim, R. Y.; Nam, M.-R.; and Kim, H. O. 2005. Fusion of multispectral and panchromatic Satellite images using the curvelet transform. *IEEE Geoscience and Remote Sensing Letters*, 2(2): 136–140.

Crampton, J. W.; Graham, M.; Poorthuis, A.; Shelton, T.; Stephens, M.; Wilson, M. W.; and Zook, M. 2013. Beyond the geotag: Situating 'big data' and leveraging the potential of the geoweb. *Cartography and Geographic Information Science*, 40(2): 130–139.

Deng, L.-J.; Vivone, G.; Jin, C.; and Chanussot, J. 2021. Detail Injection-Based Deep Convolutional Neural Networks for Pansharpening. *IEEE Transactions on Geoscience and Remote Sensing*, 59(8): 6995–7010.

Fitzner, D.; Sester, M.; Haberlandt, U.; and Rabiei, E. 2013. Rainfall estimation with a geosensor network of cars: Theoretical considerations and first results. *Photogrammetrie, Fernerkundung, Geoinformation*, 2013(2): 93–103.

Garzelli, A.; and Nencini, F. 2009. Hypercomplex Quality Assessment of Multi/Hyperspectral Images. *IEEE Geoscience and Remote Sensing Letters*, 6(4): 662–665.

Gretton, A.; Borgwardt, K. M.; Rasch, M. J.; Schölkopf, B.; and Smola, A. 2012. A kernel two-sample test. *Journal of Machine Learning Research*, 13: 723–773.

He, Y.; Zhong, Y.; and Ma, A. 2019. A deep spatial-spectral network for pansharpening. *Remote Sensing*, 11(17): 2061.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; and Chen, W. 2021. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv preprint arXiv:2106.09685*.

Huang, J.; Chen, H.; Ren, J.; Peng, S.; and Deng, L. 2025a. A General Adaptive Dual-level Weighting Mechanism for Remote Sensing Pansharpening. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 7447–7456.

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

Li, X.; Liu, J.; Chen, Z.; Zou, Y.; Ma, L.; Fan, X.; and Liu, R. 2024a. Contourlet residual for prompt learning enhanced infrared image super-resolution. In *European Conference on Computer Vision*, 270–288. Springer.

Li, X.; Wang, Z.; Zou, Y.; Chen, Z.; Ma, J.; Jiang, Z.; Ma, L.; and Liu, J. 2025. Difiisr: A diffusion model with gradient guidance for infrared image super-resolution. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 7534–7544.

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

Masi, G.; Cozzolino, D.; Verdoliva, L.; and Scarpa, G. 2016. Pansharpening by Convolutional Neural Networks. *Remote Sensing*, 8(7).

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*, 61: 1–17.

Otazu, X.; Gonzalez-Audicana, M.; Fors, O.; and Nunez, J. 2005. Introduction of sensor spectral response into image fusion methods. Application to wavelet-based methods. *IEEE Transactions on Geoscience and Remote Sensing*, 43(10): 2376–2385.

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*, MM '23, 3219–3227. New York, NY, USA: Association for Computing Machinery. ISBN 9798400701085.

Rui, X.; Cao, X.; Pang, L.; Zhu, Z.; Yue, Z.; and Meng, D. 2024. Unsupervised hyperspectral pansharpening via low-rank diffusion model. *Information Fusion*, 107: 102325.

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.; Alparone, L.; Chanussot, J.; Dalla Mura, M.; Garzelli, A.; Licciardi, G. A.; Restaino, R.; and Wald, L. 2015. A Critical Comparison Among Pansharpening Algorithms. *IEEE Transactions on Geoscience and Remote Sensing*, 53(5): 2565–2586.

Vivone, G.; Restaino, R.; and Chanussot, J. 2018. A Regression-Based High-Pass Modulation Pansharpening Approach. *IEEE Transactions on Geoscience and Remote Sensing*, 56(2): 984–996.

Wald, L. 2002. *Data Fusion: Definitions and Architectures : Fusion of Images of Different Spatial Resolutions*. Presses de l'École des Mines. ISBN 9782911762383.

Wu, R.; Zhang, Z.; Deng, S.; Duan, Y.; and Deng, L.-J. 2025. Panadapter: Two-stage fine-tuning with spatial-spectral priors injecting for pansharpening. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 8450–8459.

Yang, G.; Cao, X.; Xiao, W.; Zhou, M.; Liu, A.; Chen, X.; and Meng, D. 2023. PanFlowNet: A Flow-Based Deep Network for Pan-Sharpening. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 16857–16867.

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

Yuhas, R. H.; Goetz, A. F. H.; and Boardman, J. W. 1992. Discrimination among semi-arid landscape endmembers using the Spectral Angle Mapper (SAM) algorithm. In *Summaries of the Third Annual JPL Airborne Geoscience Workshop. Volume 1: AVIRIS Workshop*.

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.

Zhong, Y.; Wu, X.; Cao, Z.; Dou, H.-X.; and Deng, L.-J. 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.

Zhou, J. T.; Civco, D. L.; and Silander, J. A. 1998. A wavelet transform method to merge Landsat TM and SPOT panchromatic data. *International Journal of Remote Sensing*, 19: 743–757.

Zi-Han Cao, L.-J. D. G. V., Yu-Jie Liang. 2025. An Efficient Image Fusion Network Exploiting Unifying Language and Mask Guidance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.